A Data Mining Approach to Improve the Automated Quality of Data

Nawaf Abdullah Alkharboush
School of Electrical Engineering and Computer Science
Queensland University of Technology

A thesis submitted for the degree of
Doctor of Philosophy
25/November/2013

Abstract

The quality of the data in organisational information systems is a critical issue for organisations. The rapid growth in the volume and complexities of technologies related to databases and data warehouses has enabled executives and decision makers to more readily store, access, analyse and retrieve massive amounts of information to support business needs. For the most part, advantages, such as significantly improved organisational capacity, increased performance efficiency and customer satisfaction, as well as reduced operational times and costs have been the principal result. However, despite these obvious benefits, a major challenge remains, obstructing the proper delivery of these goals; this is the data quality. It is estimated that the immediate cost of a 1-5% error in data is approximately 10% of revenue. Poor data quality also has a severe impact on customer satisfaction, operating costs, effective decision making, and strategy execution.

Data quality affects many applications that facilitate data mining, database and business management. Such applications contain many interesting algorithms and techniques with which to approach the multi-dimensional problems that impact on data quality. While these methods have contributed somewhat to solving some data quality problems, they have serious limitations particularly for mining outlier data. Mining massive volumes of data to expose infrequent outlier values

based on traditional data mining tasks, such as association rule is computationally expensive. The problems that have arisen are that traditional data mining tasks are mainly designed to manage frequent data, and the focus of most data mining research is on post-process tasks; in which improving the accuracy of the data mining algorithms is desirable. Such solutions are inadequate and inapplicable when it comes to providing continuous automated solutions for detecting outlier data.

This research describes the development of a robust and novel prototype to address the quality problems that relate to the dimension of outlier data. It thoroughly investigates the associated problems with regards to detecting, assessing and determining the severity of the problem of outlier data. To address these problems, the study proposes new techniques based on granule mining, as an alternative to association rule mining, which is based on decision table theory. The proposed method is innovative and significant and has the potential to reduce the time and the costs associated with mining and assessing outlier data. Substantial experiments have demonstrated that the proposed algorithms for outlier data outperform state-of-the-art algorithms. The contribution of the experiments described herein will be to enable organisations to effectively detect outlier data and thereby assess the behaviour of data quality in a continuous automated or (a semi automated) way.

I would like to dedicate this thesis to my loving family.

Keywords

Data Mining, Granule Mining, Data Quality, Quality Assessment, Outlier Detection, Noise Detection, Data Cleaning

Acknowledgements

I would like to express my very great appreciation to Professor Yuefeng Li , for his guidance, patience, enthusiastic, and encouragement. His willingness to give his time so generously with warming friendship has significantly helped me to deliver this project. I also would like to thank Associate Professor Richi Nayak for her comments and advice. Finally, I wish to thank my colleagues for their support and encouragement throughout my study.

STATEMENT OF ORIGINAL AUTHORSHIP

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature

QUT Verified Signature

Date

25/11/2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Organisations increasingly rely on data storage technologies, and therefore data quality is becoming a critical concern of organisations. The rapid growth in the technological storage of data has brought significant advantages in terms of the capability to store massive amounts of data, the ability to retrieve relative information, and also to access heterogeneous resources. These benefits facilitate the growth of organisational locations, strategies and customers. Decision makers can utilise the more readily available data to maximise customer satisfaction and profits, and predict potential opportunities and risks. Unfortunately, this improvement in storage technologies has been at the cost of maintaining high data quality. Large quantities of poor data are saved in information systems; causing serious moral and financial implications for organisational performance.

The majority of organisational databases and data warehouses are pervaded with poor quality data Li and Joshi [2012]; Wang et al. [1995]. The appearance

of such poor quality data and the presence of various errors across databases and data warehouses significantly reduce the usability and creditability of the information systems and can have a moral and financial impact on the members of the organisation its associated stakeholders. For example, managers are unable to rely on their information systems to execute the organisational strategies they have designed; and, in some cases, conflict between employees and customers may emerge due to poor data quality. This failure is most noticeable in high rate amongst customers who choose to turn to superior service providers. These moral impacts lead to an increase in operational costs and a decrease in revenues. Studies show that the estimated immediate cost stemming from a 1-5% error rate is approximately 10% of revenue Redman [1998]. In the US, a survey conducted by The Data Warehouse Institute revealed that data quality problems cost US businesses 611 billion dollar a year Eckerson [2002].

Generally, data quality is defined as data that is fit for its intended use by consumers Ballou and Pazer [1985]. This means that, the quality of the data in a database must reflect the actual entity in the real world that is being described. When this is the case, it can be asserted that there is no poor data quality stored in the database. In the literature, there are a number of the data quality dimensions that relate to specific problems that affect data Ballou and Pazer [1985]; Lee et al. [2002]; Redman [1996]; Strong et al. [1997]; Wand and Wang [1996]; Wang and Strong [1996]. The following are the most regularly mentioned and the most critical dimensions of data quality:

- *Consistency* means that there is no conflict in data values. Conversely, inconsistency (or outliers in the data mining context) presents values that are out of range for the remainder of the collection.

- *Accuracy* refers to the value that is nearest to the value in the standard domain. Any new datum is compared against standard domain datum, which is considered to be accurate (or correct), so as to determine the accuracy of the new datum.

- *Currency* (of which timeliness and freshness are aspects) confirms the status of the data value as up-to-date. This dimension can have some influence on the accuracy of a decision that is being taken. For instance, when this dimension is neglected in information systems, a resident might receive post from commercial and governmental organisations intended for previous residents.

- *Completeness* refers to the extent to which absent or missing (or unknown) values are present in a data collection.

The above four data quality dimensions are essential factors in determining the success of many applications; including business process management, database and data warehouses, statistics, and data mining. Each of the applications developed to handle data proposes a number of interesting and promising solutions to address each of the above data quality dimensions. Based on the techniques used when approaching data quality dimensions, the literature calcifies data quality solutions into two fundamental areas: process-oriented and data-oriented techniques Batini et al. [2009]; Besiki et al. [2007]; Lee et al. [2002]; Madnick et al. [2009].

The process-oriented methods allows for the identification of the causes of data errors through observation of the whole process, since data gathering and acquisition continues until the final stage of the organisational processes. It is widely

accepted that the assessment of, and improvements in, organisational information systems cannot be achieved independently of the users perspective. Indeed, data consumers play significant roles in providing comprehensive and contiguous quality assessment and improvement. If we take the example of an organisation engaged in manufacturing fabric; if the bolt produced is expected to be one inch wide by two inches long, then, that bolt and every other bolt in the manufacturing process must also be one inch wide by two inches long to meet the product specifications Silvers [2008]. If this were not the case, the products would be different sizes and would not meet the customers' needs. The authors Dasu and Johnson [2003]; Silvers [2008] emphasise the value of improving those processes that are implicated in data quality problems, rather than aiming to fix problems inside the database when they manifest at a later stage. This also ensures that the manufacturing process is under control and will incorporate savings of both time and cost.

However, it is inefficient to rely solely on process-oriented methods to handle data quality problems for numerous reasons. The first one is that implementing a process-oriented method is a difficult and time consuming task, requiring frequent process checking of all organisational processes and sub-processes. Hence, massive losses in time are associated with pursuing process-oriented approaches to deliver data quality. Additionally, process-oriented methods are subjective, and largely relate to the user's decision. Reaching a consensus can be difficult in such cases, particularly when dealing with data quality dimensions such as inconsistency (or outlier data). More importantly, the ultimate decision as to whether or not process-oriented methods ensure data quality in a database or a data warehouse is determined by data-oriented solutions.

Therefore, the techniques invoked for the purpose of data-oriented solutions play an essential role in insuring data quality. There are significant contributions discussed in the literature reviewed in this work, that involve data quality problems and dimensions from a data-oriented perspective. Several approaches that adopt data-oriented methods address different data quality issues, such as records duplication, incomplete, inaccurate, and inconsistent data. Thus, it is apparent that users can utilise data-oriented techniques to objectively assess, improve, and maintain the quality of the data held in their databases and data warehouses. The data-oriented view also enhances the process-oriented view, as it enables data quality experts to benchmark quality change, allocate process locations that generate quality problems for processes re-engineering, and also reduce the time frames and resources needed to conduct a complete object data quality study to assess data quality dimensions.

Due to the breadth and the dimensionalities of data quality research, this research expands exploration of the problem to the dimension of outlier (or inconsistent) data from the perspective of the data view. The outlier problem is a critical one toward many applications including fraud detection, network intrusion, and the monitoring of terrorist activities and healthcare. The data in these applications can either be stored in a single database, multiple databases or a data warehouse. Hence, mining outlier data from each of these three types is difficult and becomes increasingly challenging in cases where there are multiple databases and data warehouses.

In the case of the database, there are two types of the databases in most organisations: a single data source and multiple data sources. This will depend on the size of the organisation and the services it provides. In a single data source

the complete solution for outlier data must be considered at both the schema and data levels; thus, users need to clearly and correctly define, analyse, assess and improve the outlier data at both levels. This can be difficult because of the massive data throughput that experts need to deal with.

In cases with multiple databases, outlier detection becomes even more tedious, expensive and time consuming. The reasons for this are associated with multiple schemas and data levels. Data quality experts need to individually assign specific quality conditions and constraints for each of these multiple data sources in such a way as to avoid conflict. This can be difficult to establish and generalise because some rules differ at the schema level or/and the in data level when considering outliers in a databases, as compared to normal data. Having single or/and multiples databases pervaded with outlier data can cause a great challenge when building a historical reliable decision support system in a data warehouse.

A data warehouse is a historical repository consolidating multiple databases from different periods of time. There are three steps that must be undertaken when designing a data warehouse. These steps are extract, transfer and load; commonly referred to as the ETL process. The successful construction of a data warehouse is highly reliant on the quality of the databases whose data will be extracted and transformed and loaded into the data warehouse. Although, some ETL tools provide quality capabilities for transforming steps to include some business rules, clean constraints and integrity, these quality capabilities cannot guarantee highly consistent data quality in a data warehouse if the quality of the data in the original databases contains outlier values.

With such massive data sets in the database and the data warehouse, there is

growing attention being paid to Knowledge discovery in database (KDD); something which is commonly referred to as Data mining (DM). DM is useful when dealing with very large data sources. DM provides various techniques that enables users to extract interesting patterns, find associations and retrieve valuable information. Given the benefits of Data Mining techniques, they are used widely for information retrieval, fraud detection, medicine and healthcare, network security and data quality.

KDD or DM consists of several components. The first and foremost component in any data mining project is that the data quality of the dataset should be high. The quality of the dataset can potentially impact on the remainder of the phases in the data mining process. It is estimated that up to 60% of the time allocated to a data mining project is consumed in the preprocessing stage, which includes data cleaning, normalisation, transformation, feature extraction and selection. Data cleaning tasks are at once the most tedious and the most critical task for two reasons. Failure to provide high data quality in the preprocessing stage will significantly reduce the accuracy of any data mining project. Hence, the phrase "garbage in garbage out " becomes true in the case of a data mining project.

DM literature provides various techniques and algorithms with which improve data quality in the preprocessing stage. These algorithms utilise different data mining techniques to capture and improve different quality problems that might be associated with the data. However, the main limitation of these studies is that data quality issues are investigated for the purpose of improving the accuracy of the post-processing for the purposes of clustering, classification and association rules algorithms. This somewhat narrow view of the problem disregards the

necessity for continuous data cleaning and the maintaining of high data quality in databases and data warehouses and can lead to outlier values occurring in the system. Hence, users cannot rely on these approaches to provide an ultimate solution to guarantee data quality.

Over the years, there has been growing understanding of the necessity to reflect on data mining techniques to support the automation of data quality in source systems: database and data warehouse. A plethora of data mining techniques and algorithms have been developed, but their remains a gap, particularly in reference to the dimension of outlier data and the assessment of quality change transformation in reference to outlier behaviour from time to time. It is intended that the contribution made by this study will go some way to filling this gap.

This thesis will detail the extensive investigation which was undertaken to overcome the challenges of providing a complete automated solution for outlier data. The main role of this study has been to propose new and promising algorithms for detecting outlier data. Additionally, this study acknowledges that poor data, such as outlier data, can systematically or randomly appear across columns and rows with variance in degrees. Hence, the work described in this study enhances the proposed outlier algorithms with new quality assessment measurement that will enable users to assess the quality change in outlier behaviour periodically. The substantial experiments in this study demonstrate the promising contributions of the proposed outlier algorithms and new quality assessment measurements.

1.2 Problem Statement

The significant improvement in the size and technology of information systems has increased the necessity for data mining techniques and algorithms in many applications; in particular to ensure data quality. The reasons for this are associated with the capability of data mining techniques and algorithms to efficiently and effectively scale well with large datasets, extract frequent patterns, find association rules, classify information and predict future opportunities and risks. These advantages enable researchers to design semi and fully automated approaches that can handle different data quality problems including: records duplication, record linkage and various data quality dimensions.

However, there are some problems present in the traditional data mining techniques utilised for mining outlier data, because outlier data appears infrequently in the system. It is formally defined as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism " Hawkins [1980]. Other domains of knowledge define outliers as sets of examples that are a far distant from their neighbours Knorr and Ng [1999]; Ramaswamy et al. [2000]. Another assumption held in reference to outlier data is derived from Frequent Itemsets FIs where outliers are the points which have infrequent common patterns. The above definitions of outlier data lead us to highlight the probability of the following problems arising in the current research, when seeking to provide a complete solution for outlier data:

- Most traditional data mining techniques such as classification, clustering, frequent itemsets and association rules are designed for dealing with frequent data. Hence, utilising these techniques for mining massive data for

outlier detection is difficult and computationally expensive, as the outlier data infrequently appears in databases and data warehouses. Although, the literature provides a wide range of data mining methods that deals with infrequent and exposed outlier examples, these methods to some extent are computationally expensive and inapplicable when considered in reference to increasing data volumes and dimensionalities, or when restricted to a numeric data type.

- Existing outlier detection methods are only focused on one aspect of the data quality problem which is detecting outlier data. Other essential aspects of data quality such as quality assessment and improvement are not well defined in either data mining or outlier literature. Considering these aspects: outlier detection and outlier assessment are essential to provide a semi and fully automated data quality solution to classify outlier data in a database and data warehouse.

- Literature provides various outlier detection methods which can be classified into two approaches: parametric and non-parametric. The statistical method, also called the parametric method assumes that data is parametric or normally distributed. The key problem when adopting statistical approaches is that most KDD applications are multidimensional with unknown underlying data distribution.

- The non-parametric category overcomes the limitations of parametric methods and is more promising for effective outlier detection. The non parametric method was first introduced in papers Knorr and Ng [1998, 1999]. The studies Knorr and Ng [1998, 1999] introduced a non-parametric method

called the distance-based approach. The outlier in a distance-based is defined based on its distance to other examples. The main advantage of distance-based approach is such that the user does not require any prior knowledge of the data distribution. Other popular non parametric methods for outlier detection are density-based and Frequent Itemsets FIs. However, the distance-based and density-based methods require quadratic computational complexity with respect to data size and dimensionalities.

- Frequent itemsets for outlier detection scales work well with increasing data size and dimensionalities. Yet, it is difficult to specify a minimum support threshold which separates frequent items from infrequent ones. The reason for this is that if the specified threshold is large, then we are likely to have a large number of outlier candidates. Vice versa, if the specified threshold is small, then we may miss some outliers. Additionally, the accuracy of the frequent items might fluctuate when changing the minimum support as the size of the items would change.

- Another significant problem, besides detection of outlier data that has not received much attention, is how best to assess the data quality of outlier data periodically. Most data quality research is often compromised by the presence of poor data dimensions such as outlier data. Detecting outlier data is undoubtedly an essential step for data mining as it could influence the accuracy of mining tasks such as classification or clustering. Yet, this limited view of the problem does not guarantee a complete outlier solution. Outlier data is likely to continuously occur in a database and a data warehouse. The existing solutions for quality assessment do not support users

to clearly and accurately assess the change in outlier behaviour periodically, or enable them to improve the database by eliminating any malicious behaviour caused by access to the information systems.

- The current approaches to quality assessment are primarily dependent on manual inspection with little support for automated techniques. Most quality assessment methods rely on the error rate or accuracy rate to expose the ratio of outlier data in the database or data warehouse. These methods are likely to provide misleading results when users want to compare outliers between databases across different time periods; this is because as distribution of outlier data might differ between these databases.

- Quality assessment methods omit mention of the fact that outlier data can be systematically and randomly distributed across columns and rows. Hence, users are unable to learn from old systems so as to allocate the appropriate time and technological resources to improve the quality of their systems when capturing suspicious outlier data prior to accessing their systems.

To address these limitations in both outlier detection and quality assessment, it is important to solve the challenge associated with both outlier detection and assessment methods and also to incorporate quality assessment into outlier data so that newly generated outliers are efficiently and effectively captured prior to accessing the information systems. Additionally, decision makers can clearly allocate the most severely affected data, determine patterns and estimate the time and complexity required to conduct quality improvement tasks. All this can undoubtedly make a breakthrough towards the automation of quality improvement

in outlier data.

1.3 Contributions

The majority of existing methods in data quality research are focused on one aspect of data quality which is detecting poor data quality. Due to this limited view of data quality problems, there are no efficient and effective solutions that can be said to represent a breakthrough towards an automated data quality solution. This thesis thoroughly investigates and discusses these limitations in Section 6.2; particularly in the dimension of outlier data and significant contributions toward proposing a complete automated data quality solution. It also considers two essential aspects of data quality: Outlier detection for exposing outlier data and quality assessment for assessing any quality change in outlier behaviour periodically. As a consequence its significant contributions are anticipated to be the following:

- It will draw attention to aspects of data quality in order to facilitate a complete automated data quality solution. Particularly, the thesis will investigate the limitations of existing outlier data capture with regards to large datasets. The thesis will also study the limitations of existing data quality and outlier methods by providing complete automated solutions for outlier data.

- Practically, the thesis will propose several algorithms for mining outlier data: categorical and mixed attributes datasets, by utilising the concept of rough set theory. Initially, the thesis will introduce two approximation al-

gorithms: the Decision table DT and the Weighted decision table WDT for finding and approximating the location of possible outlier patterns. These DT and WDT algorithms have great advantages for mining outlier data, as the number of patterns found by DT and WDT is significantly smaller than that of other methods that rely on frequent items sets found by an Apriori algorithm.

- This research will contribute by suggesting three outlier algorithms for mining outlier data. The proposed algorithms will be effective for mining outlier data because of the mining space found in the approximating algorithms: DT and WDT, is very small. The first outlier algorithm; the GBOD, is based on an approximation of the DT. The remaining algorithms are a ranking weighted decision table RWDT and centroid granules CG are derived from an approximation of the WDT algorithm.

- Additionally, the study understands the limitations of existing data quality with regards to quality assessment. Hence, the thesis will contribute by assessing outlier data using two algorithms. The first algorithm for quality assessment measures the quality change of outlier data from different time periods. The second algorithm for quality assessment captures the root cause of outlier data and determines the location of the most severe outlier data by measuring the systematic and random degree of outlier data.

- Substantial experiments and analysis of all the techniques and algorithms proposed in this thesis are undertaken so as to clearly validate their effectiveness; not only for exposing outlier data but also for assessing quality transformations in the outlier behaviour periodically.

1.4 Dissertation structure

To ensure a comprehensive coverage of the research problems, existing methods and the proposed goals, this thesis is organised sequentially as follows:

- **Chapter 2:** provides comprehensive coverage of related data quality literature. The critical review of the literature in this chapter clearly exposes a gap in the existing data quality research.

- **Chapter 3:** details the framework for automated data quality in this thesis. The framework can be considered as a roadmap for the thesis's contributions. It shows the flow, the contributions of the work and associated chapters.

- **Chapter 4:** is mainly focused on extracting useful patterns. Particularly, the chapter introduces two useful methods. The first one is based on rough set theory. The second one is a novel approach based on using weighted decision table WDT. The advantages associated with these two methods lend critical success factors to the remaining contributions with regard outlier detection and outlier assessment.

- **Chapter 5:** addresses the limitations of existing outlier algorithms. It introduces three outlier algorithms to address different outlier problem. The first algorithm is the GBOD, which demonstrates how use of the Discernibility Matrix can provide accurate outlier information in the form of traditional Euclidean distance. The RWDT algorithm provides efficient ranking outlier data. The third outlier algorithm (Centroid granule) uses centroid granules. The centroid granules algorithm measures the distance between a

pattern and a centroid pattern rather than between a pattern and a number of nearest neighbour patterns, to determine its outlier degree. The proposed WDT and RWDT for outlier detection are accepted as journal paper by International Journal of Intelligence Science(IJIS) .

  - Nawaf Alkharboush , Yuefeng Li and Richi Nayak. 2012. Outlier Detection with Weighted Granule Mining.

- **Chapter 6:** assess the outlier data and measures its severity. The algorithms proposed in this chapter represent a major breakthrough towards the automation of data quality assessment. The first algorithms enable users to measure the quality of change with regards to outlier data. The second contribution in this chapter investigates the systematic and random distributions of outlier data in database. The relevant publications of the proposed quality assessment are as below:

  - Nawaf Alkharboush and Yuefeng Li. 2010. A Decision Rule Method for Assessing the Completeness and Consistency of a Data Warehouse. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent.
  - Nawaf Alkharboush and Yuefeng Li. 2010. A Decision Rule Method for Data Quality Assessment. In Proceedings of the 2010 ICIQ/ACM International Conference on Information Quality.
  - Nawaf Alkharboush and Yuefeng Li. 2011.A Decision Table Method for Randomness Measurement. In Proceedings of the 4th International Conference on Intelligent Decision Technologies.

- **Chapter 7:** includes substantial experiments, analysis of the proposed techniques and algorithms in this thesis. For the purpose of fair validations, this thesis conducts extensive experimental studies and compares their results to several state-of-the-art algorithms, to determine the effectiveness of the proposed algorithms to address research problems.

- **Chapter 8:** is the conclusion to the thesis. It also suggests possible future research directions.

Chapter 2

Literature Review

2.1 The Concept Map of Literature

This chapter presents a review of the current literature that discusses data mining for data quality, particularly that which focuses on the dimension of outlier data. To ensure complete coverage of relevant literature, the literature review has been constructed around four major topic areas as in Figure 2.1. The first section covers the issues associated with data quality and discusses the quality issues that arise as a consequence of existing methods used for databases and data warehouses. The second section of the literature reviews the knowledge discovery process for database and data mining tasks. Section three of the literature review concerns data mining; particularly emphasising identification of frequent patterns and the extraction of knowledge using association rule mining. This section covers the decision table and rule generation based on a rough set theory. The final part of the chapter thoroughly discusses the quality problems that relate to the dimension of outlier data, since the principal goal of the thesis is to

Figure 2.1: Literature Review for Data Quality in Data Mining

provide a complete automated solution to guarantee data quality with regards to outlier data. It will cover practices associated with quality programming, so as to describe related outlier literature. Specifically, this section includes a definition of outlier data, reflects on existing outlier detection methods, and assessment approaches to outlier data.

The research and studies presented throughout this chapter suggest that there is a significant gap in existing conceptions, algorithms and methods in relation to providing automated data quality solutions with regards to outlier data. It is this gap that the this study seeks to fill.

2.2 Data Quality

Data quality is a critical factor associated with technological solutions to manage data in support of organisational performance. It can be defined as fit for use by data consumers Ballou and Pazer [1985]. In other words, data must be stored at a high quality to reflect consumers' needs. However, in a real world setting, organisational information systems are pervaded with poor data quality. A survey of 500 medium-size firms with annual sales of more than $20 million confirms that 60% of participating firms experience data quality problems Wand and Wang [1996]. The emergence of poor data can have a severe impact on decision making processes and customers' and employees' satisfaction levels, as well as on making and executing strategies and increasing operational costs Redman [1998].

Data quality research involves a series of steps, including Defining, Measuring, Analysing, and Improving data quality. The procedure followed to test quality is advocated by leading quality programmers such as MIT Total Data Quality

Management (TDQM) and the Department of Defence (DoD) Lee et al. [2002]; Redman [1996]. These four components are gradually and seamlessly interrelated with one another. To meet the aim of achieving a suitable and comprehensive data quality testing method the users need to incorporate these four phases into any data quality framework. For instance, management cannot improve systems if they are unable to define or measure defective values. Neglecting any single phase can have a potentially severe impact on data quality evaluation.

Researchers and practitioners investigating the four data quality components stated above have produced studies that make outstanding theoretical and practical contributions to this field. These studies have resulted in the division of data quality research according two fundamental viewpoints:

- Process-oriented
- Data-oriented

Both perspectives have been conceptualised as a data manufacturing system wherein data is interpreted in the same way as manufactured product Ballou et al. [1998]; Wang and Kon [1993]; Wang et al. [1995]. Both can incorporate a subjective (User perspective) or/and an objective (data-oriented perspective) assessment of each item of data in all quality analysis phases: Defining, Measuring, Analysing, and Improving. Process and data views are intrinsic benchmarks against which quality can be improved Redman [1996]. A proper implementation of these views is intended to ensure continuous quality improvement for both organisational processes and data sources.

Significant contributions made in data quality research proceed from the process-oriented viewpoint. There are three main roles in the data production process: data producers who are in charge of the process in which data is generated; data custodians who take responsibility for providing, managing and maintaining data storage; and data consumers (people or groups who use data) Strong et al. [1997]. Authors support the process view, arguing that conducting data quality research by defining, measuring, assessing and improving data quality dimensions directly from a database and/or a data warehouse are insufficient solutions when it comes to ensuring uniformly high data quality Dasu and Johnson [2003]; Redman [1996]; Silvers [2008]. Their main argument is that errors are likely to arise and so be stored again in data source systems. Therefore, the adoption of a process view enables manages and decision makers to identify and understand the root causes of quality problems Dasu and Johnson [2003]; Redman [1996]; Silvers [2008] and therefore ensure a continuously high quality of data. Data quality literature includes several approaches that address different quality dimensions based on a production process view Lee et al. [2002]; Strong et al. [1997]; Wang and Strong [1996].

However, it is crucial to note the several drawbacks that are associated with a process-oriented view. The major drawback being that decisions made regarding quality issues are subject to data custodians' opinions. This could cause conflict to arise between custodians when determining whether or not a nominated case is experiencing a quality problem. The level of conflict among custodians is dramatically increased, and can also become out of control, in response to the complexity and dimensionalities of data quality problems. Moreover, the ultimate decision as to the quality of a process view is highly dependent on measuring and checking

quality in systems sources: i.e. the database and the data warehouse. Users who are using a process view need to measure different data quality dimensions including: outliers, completeness, currency and accuracy. An additional drawback is the complexity and time consuming nature of applying a process view, because data quality measurement is a continuous procedure involving several continuum phases. Thus, restrictions apply when solving quality issues using a process view only, as it is not possible to guarantee continuous and sufficiently regular quality monitoring and maintenance for large database and data warehouses. Finally, the process view relies on human involvement to deliver data quality solutions; again this increases costs incurred in terms of both time money and makes quality control a tedious procedure.

There are various statistical and data mining approaches to objectively deliver data quality solutions. These methods can potentially benefit the process view by examining quality problems, identifying common patterns among defective data, benchmarking of organisational processes and recommending appropriate solutions. To this end, the literature reveals that increased attention has been paid towards data quality research from the perspective of a data-oriented view. The main motivation to utilise a data-oriented view to develop data quality solutions is that the most severe quality issues are pertinent to the values of the data. Thus, a proper method of data quality measurement from a data-oriented perspective is required. Also, a data-oriented view offers solutions to reduce and illuminate the involvement of manual inspection over poor data, as data-oriented methods utilise semi and fully automated techniques to deal with different data quality problems. Hence, a data-oriented view has great advantages compared with the process-oriented view, especially with time and cost efficiency, but also in terms

of the possibility for continuous monitoring and for controlling the quality in a database and data warehouse.

Despite the advantages of employing a data-oriented view, the literature to date is limited, in that it mainly focuses on the detection and correction of poor data. A comprehensive classification of dirty data based on data view presents in Kim et al. [2003]. Study in Kim et al. [2003] investigates the appearance and manifestation of different dirty data in a database and data warehouse. The authors present a taxonomy structure to describe different types of dirty data and identify the causes of these data. The authors also present another taxonomy that describes possible solutions for each item or set of dirty data. However, to the ability to detect and correct dirty data is not a solution to the requirement to provide automated data quality solutions. Users cannot rely on these methods to capture the root causes of problems, nor can they apply them to measure the quality of changes in the database.

2.2.1 Data Quality in a Database

There are many sources that increase the data quality problems that occur in a relational database. Paper Kim et al. [2003] illustrates different categories of dirty data and describes the reasons that cause them to arise. These causes can be grouped into two fundamental elements. The first element is associated with human activities: Users generate different types of errors by deliberately or mistakenly entering incorrect, outlier or incomplete data. The second element is associated with a poor and improperly designed original database. The design of a database can positively or negativity impact on the quality of the database

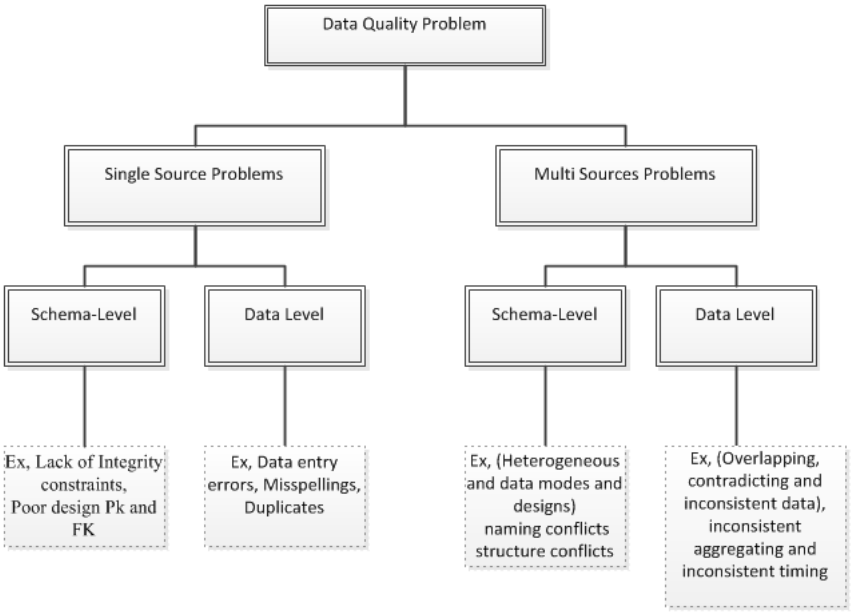

Figure 2.2: Quality Problems in a Database

as it evolves. Specifying constraints in the schema level of the database plays a major role in addressing quality issues such as incomplete or outlier data.

A Study in Rahm and Do [2000] classifies quality problems in a relational database into two areas: Single source problems and Multi source problems. Within each area, quality problems appear in both schema and instance levels. Figure 2.2 illustrates quality issues in the relational databases from single and multi sources.

Paper Rahm and Do [2000] goes on to define the quality problems in single and multiple databases as follows:

Single-Source Problems

- Schema Level problems: occur because of a lack of appropriate model specific or application specific integrity constraints.

- Instance Level problems: relate to errors and inconsistencies, such as misspellings, that cannot be avoided at the schema level.

Multi-Source Problems

- Schema Level problems: are caused by naming conflicts; involving either using the same name for different objects, or using different names for the same objects. In addition, structural conflicts can occur with many variations, referring to different representations of the same object in different sources, for instance, attribute vs. table representation, different component structure, different data types, different integrity constraints, etc.

- Instance Level problems: reflect data conflicts, which means that data is represented differently in different sources. Moreover, even when data has the same attribute name or data type, there may be value differences.

It becomes more difficult to address quality at the instance or schema levels, particularly when the user is dealing with dimensions of outlier data. At the instance level, users either need to rely on domain experts or automated outlier detection methods to detect outlying values. Detecting outlier values by relying on domain experts is not an efficient solution due to the vast volume of data. However, automated outlier detection methods also encounter some difficulties with correctly flagging outlier records and outlier values, in such a way as to efficiently scale with large dimensional databases; as discussed in Section 2.5.

The quality of a schema mostly relies on setting different constraints. The use of constraints enforces users to follow rules associated with these constraints, and therefore guarantees that no data breaches the constraints stored in the database.

For example, in the incomplete data quality dimension, a database designer can set a constraint that enforces users to not leave missing fields. This ensures that no missing values appear in the database. However, specifying constraints for outlier data is difficult to incorporate into a design. The reason being that the database designer typically has no foreknowledge of what outlier values might emerge or be expected.

The only possible way to guarantee prevention of outlier data is through the use of triggers. Triggers are sets of rules that are generated from the outlier data itself. Hence, experts can initially mine a database to detect outlying values from the instance level. After outlier values are exposed, experts investigate the association between these and existing attributes in order to generate rules that prevent new throughput outliers; this reduces the efficiency of the cleaning system because the system needs to verify every new item of data with a set of trigger rules. Although, the trigger is essential to prevent outlier values, users need to find an innovative approach to detect and find the association(s) among outlier records.

2.2.2 Data Quality in a Data Warehouse

Data warehousing is one of the most promising technologies used to assist middle and senior management teams. It provides details of historical information for business needs to enhance the decision making process Elmasri and Navathe [2007]. There is no unifying definition of a data warehouse as it is normally something developed by numerous organisations to meet their business requirements. However, generally a data warehouse can be defined as a "store of integrated data

from multiple sources, processed for storage in multidimensional model " Elmasri and Navathe [2007].

When designing a data warehouse the developer is heavily depend on the ETL (extraction, transformation and loading) processes. ETL processes are responsible for performing essential steps that include: extraction data from heterogeneous sources, cleansing tasks, customisation and loading data into a data warehouse Vassiliadis et al. [2001]. These activities involve significant operational challenges and complexities. Figure 2.3Chaudhuri and Dayal [1997] describes the processes of designing a data warehouse and the task of implementing the ETL processes. The literature provides various models to deal with complexity, usability and the price of ETL tools. Papers Vassiliadis et al. [2000, 2001] present a uniform meta model for ETL processes, activity modelling, contingency treatment and quality management. In their studies the authors of Vassiliadis et al. [2000], propose a methodological approach, describing how semantically rich meta-information in a data warehouse can be stored in a meta data repository.

However, due to the many data quality problems that have been seen to affect data warehouse projects, implementation of the ETL process is exceptionally challenging. Even though, the ETL process incorporates some data quality detection and assessment capabilities, the numbers of the constraints and specified triggers that would be required in ETL processes to control dirty data are prohibitively large. This undoubtedly increases the operational process elements of ETL systems. Additionally, the ETL process does not in any way guarantee that clean and reliable data is present in a data warehouse. Moreover, users cannot determine if the data being loaded into a data warehouse reflects the same data as that present on relational databases Silvers [2008].

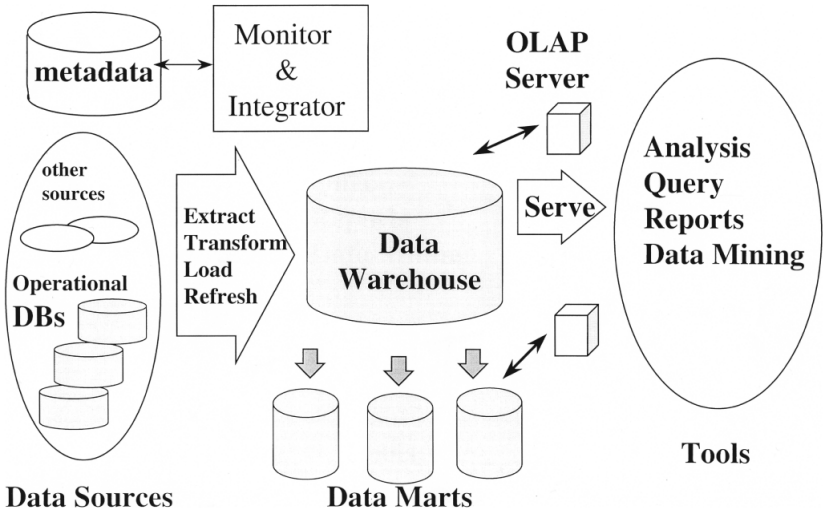

Figure 2.3: Data Warehousing Architecture

Since data warehouses typically contain large volumes of data gathered from heterogeneous sources, it is essential that they "support highly efficient cube computation techniques, access methods, and query processing techniques " Han and Kamber [2001]. The support of OLAP cube (on-line analytical processing) computation of data in the data warehouse allows users to analyse data to search for business intelligence. The OLAP cube includes operations such as slice and dice, drill down, roll up, and pivot, which support the heretical structure view of the data warehouse. Figure 2.4 illustrates a data cube consisting of three dimensions: Date, Products, City. By employing OLAP operations, executives and analysts can view information in the data cube from several details on the top level to more details when drilling down to the lowest level of the cube Chaudhuri and Dayal [1997]; Ge et al. [2003] as shown in Figure 2.4Chaudhuri and Dayal [1997].

Although, the data in the OLAP cube supports users in extracting interesting

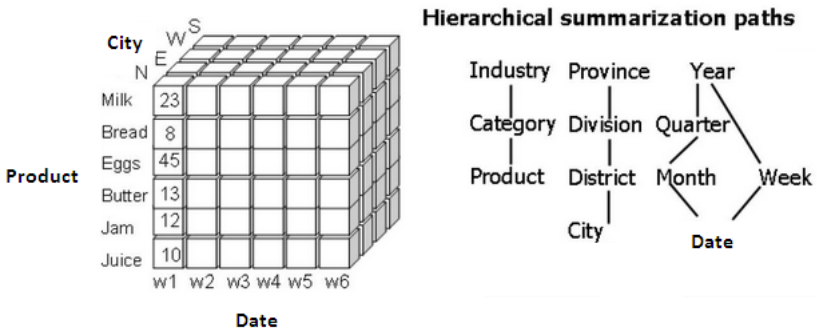

Figure 2.4: Multidimensional Data Cube Messaoud et al. [2006]

information based on multiple levels of granularity, the OLAP cube does not have the capabilities to automatically explain associations amongst data Messaoud et al. [2006]. Therefore, several data mining techniques, in conjunction with OLAP cube capability can be useful in acquiring interesting knowledge from the data warehouse Messaoud et al. [2006]. Particularly in reference to data quality, the OLAP cube and data mining techniques have brought significant advantages when seeking to automatically define, detect and improve on different data quality problems that emerge in a data warehouse Berti-Equille [2007].

Despite the contributions of these methods, particularly for dealing with outlier problems, they mainly focus on detecting outlying values for the purpose of enhancing the accuracy of the data mining algorithms, such as for classification and clustering. Because of this, users cannot rely on these studies to provide a complete automated outlier solution. To move towards automated data quality for the dimension of outlier data, experts need to take a broad view, to ensure that their systems detect outlier values, provide complete assessment of outlier behaviour and trigger outlier values. Moreover, existing outlier algorithms to some extent, creating efficacy and effectiveness problems when dealing with a

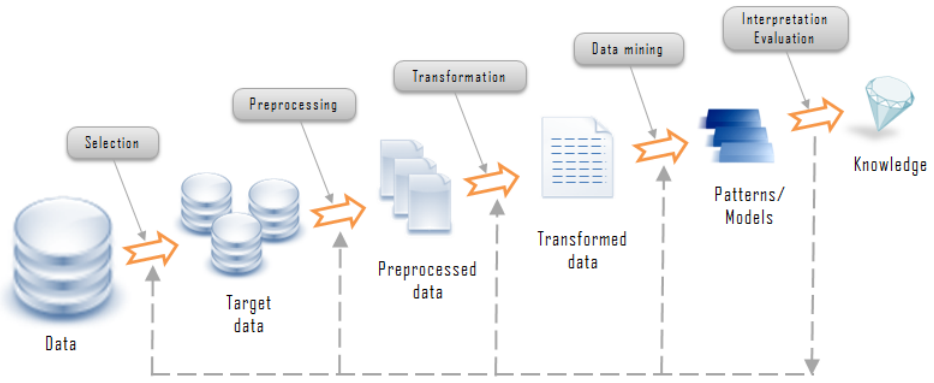

Figure 2.5: Knowledge Discovery Process Fayyad et al. [1996]

large dimensional database or data warehouse.

2.3 Knowledge Discovery in Database

The Knowledge discovery in database (KDD) is a continuum with steps that interactively and iteratively depend on each other. In real life applications, extracting useful knowledge to support business needs is a difficult task. The rapid growth in data size and technologies, as well as the availability of storing and accessing different data types includes: structured data, text data in web, images, videos increase the necessity of adopting the KDD process in Figure 2.5 from paper Fayyad et al. [1996].

2.3.1 Knowledge Discovery Processes

The Knowledge discovery process is generally classified into two areas: Preprocessing steps and Post-processing steps. The pre-processing stage includes: data cleaning, data selection and data transforming, whereas post-processing

consists of data mining, pattern evaluation and knowledge representation. These steps that occur in both pre-processing and post-processing are seamlessly related to each other. Figure 2.5 shows the phases of the knowledge discovery process, as briefly described in the following points:

- Data Cleaning: This step concerns data quality in the database and the data warehouse. Data must be checked and cleaned prior to moving it forward in the KDD process. Many quality problems are handled at this stage including: outlier or noisy data, missing fields and inaccurate data Fayyad et al. [1996].

- Data Selection: This phase is very useful for reducing the dimensionalities of the dataset. In the data selection stage, users need to select useful features to represent the data. The selection of such features varies and depends on the goal of the data mining task.

- Data Transformation: In this stage, the data is transformed and consolidated based on the specified data mining tasks. Transformation methods include: normalisation, aggregation, generational and attribute redesign, which can be used in transforming data.

- Data Mining: This stage refers to the data mining tasks that users tend to adopt in a nominated KDD project. There involve the number of data mining tasks: pattern summarisation, classification, clustering and association rule mining. Based on the data mining tasks, there are a numbers of techniques and algorithms that can be used to identify the patterns from the data. This usually results in huge and meaningless numbers of patterns.

- Pattern Evaluation (interpretation): Data mining tasks often produce an overwhelming number of meaningless patterns. Users need to evaluate and interpret these patterns to identify those interesting patterns that are relevant to the targeted application.

- Knowledge Representation: After locating interesting patterns, users need to encapsulate these patterns in knowledge. This knowledge can be incorporated and represented by users or the system in order to apply this knowledge to unseen data.

2.3.2 Data Mining Tasks

Data mining is defined as a process involving the extraction of useful and interesting information from the underlying data Han and Kamber [2001]. Based on the specific application, users can deploy a single data mining task or can combine more than one data mining tasks in order to extract useful and interesting information. Data mining tasks can be described as follows:

- Pattern summarisation: The main problem in data mining is that the total number of patterns is considerably large. Even after filtering out some of the more frequent patterns that fall over the specified minimum threshold, the number of patterns remains huge. Thus, manual examination by domain experts over the patterns is undoubtedly difficult to achieve. Therefore, it is essential to adopt pattern summarisation methods, such as the profile-based approach presented in Yan et al. [2005] to allow for significant reduction in the number of patterns.

- Classification: is a supervised data mining technique. It aims to correctly classify a set of features related to set classes. The function or the model that emerges between set features and classes in the training data can then be used to predict the classes for new data in the testing set. The accuracy of the model depends on accuracy when assigning a set of features or objects as belonging to classes Han and Kamber [2001].

- Clustering: is an unsupervised data mining technique. In clustering, instances are divided and grouped into a number of clusters based on the resemblance between instances. Those instances belonging to the same cluster share many characteristics. A classic clustering technique, which is based on K-means, involves the user initially specifying the number of desirable clusters, as K. Then, based on the ordinary Euclidean distance metric, instances are assigned to the closest clusters Han and Kamber [2001].

- Association rules mining: is one of the most powerful data mining techniques. Association rule mining was first presented in Agrawal et al. [1993] for use when mining frequent itemsets in transaction databases, and has since then been developed for the purpose of mining frequent itemsets at multiple levels Han and Fu [1995, 1999] and intertransactional itemsets Feng et al. [2002]; Tung et al. [2003]and correlations between itemsets Shichao et al. [2006]; Tsumoto and Hirano [2003] . Association rule mining includes two phases. The first phase is called pattern mining; that involves the discovery of frequent patterns. The second phase is called rule generation and involves the discovery of interesting and useful associations rules in discovered patterns. The association rule is somewhat useful for measuring

associations between itemsets.

2.4 Data Mining Techniques

2.4.1 Frequent Pattern Mining

All data mining tasks including association rule mining, classification and clustering are designed for use with frequent patterns. Frequent Patterns Mining involves:

- Frequent Item Set Mining
- Frequent Sequence Mining:
- Frequent Tree Mining:
- Frequent Graph Mining:

This literature review section of the thesis mainly focuses on frequent pattern mining as this is the type that is most related to the method proposed in this thesis. Frequent pattern mining was introduced in Agrawal et al. [1993]. The Apriori algorithm from paper Agrawal et al. [1993] involves a bottom up searching approach, which initially begins with a single item, extending to long item sets at the lowest level. Apriori algorithms consist of two steps: candidate generation and rule finding. The procedure for an Apriori algorithm is summarised as follows:

- Specifies the minimum support to distinguish between frequent and infrequent items.

- Scan over the database and compute the candidate C1 for one item and its support.
- Pruning all items in C1 that have less support than the specified threshold. threshold.
- Rescan the database to compute the candidate C2 for two frequent items and pruning all items with supports below the support threshold.
- Repeat the above step till no further candidate can be found.

However, the candidate generation process from the Apriori algorithm often takes a long time to find all frequent item sets. This step also includes additional noise Pei and Han [2002]. Studies by Han et al. [2000, 2004] focus on finding frequent items without including the step of candidate generation. Papers Han et al. [2000, 2004] propose a frequent pattern tree FP-tree structure; an FP tree based mining method and FP-growth for mining complete frequent patterns. These techniques have been shown to afford significant advantages with regards to compressing large data sets into much smaller data sets and when avoiding the cost of generating large number of candidate sets.

Yet the number of item sets and the quantity of rules generated is still large. Several studies attempt to reduce the size of frequent items by constructing condensed representations of frequent item sets. Paper Bayardo Jr [1998] presents a MaxMiner algorithm for mining maximal frequent patterns. The Max-Miner algorithm extracts the maximal frequent item sets, wherein the item set is considered as a maximal frequent if there is no frequent superset. A study from Zaki et al. [1999] introduces the CHARM algorithm for mining closed item sets. The

CHARM algorithm explores both item set spaces and also creates a set space which enables the algorithm to utilise a fast search method to identify the closed frequent item sets, rather than the many non closed subsets Zaki et al. [1999]. Other contributions on condense representation of frequent item sets proposes in Calders and Goethals [2002, 2007].

2.4.2 Association Rule Mining

Association rule mining was first introduced by Agrawal et al. [1993] in order to identify interesting relationships between items in a data set. The authors developed two algorithms, Apriori and AprioriTid, to discover all significant association rules between items Agrawal and Srikant [1994]. Association rule mining is useful because it provides the user with objective measures based on the structure of patterns and statistics. An association rule is an implication $X \rightarrow Y$, where X and Y are sets of items of transactions from a single table, relational tables, or several rows and tables in a given database. Thus, whenever a transaction contains all items $x \in X$ then this transaction is likely also to contain all y $y \in Y$ with probability Hipp et al. [2001]. The following section show the formula for computing support and confidence:

- Support is the probability of transactions that contain $X \cup Y$. This can be measured by using the following formula:

$$Support(x \rightarrow y) = \frac{\sum(X \cup Y)}{T}$$

,where $\Sigma(X \cup Y)$ is the total number of transactions that contain both $X \cup Y$ and T is the total number of transactions.

- Confidence is the conditional probability which can be computed using the following formula:

$$Confidence(x \rightarrow y) = \frac{\sum(X \cup Y)}{X}$$

where $\Sigma(X \cup Y)$ is the total number of transactions that contain both $X \cup Y$ and X is the number of transactions X.

However, generating all association rules from frequent item sets is a time consuming activity Han and Kamber [2001] and can generate many redundant rules Xu and Li [2007]; Zaki [2004]. A number of studies have endeavoured to reduce the number of rules by finding close rules Zaki et al. [1999] and deduction rules Calders and Goethals [2002, 2007]. Papers Chen et al. [2005]; Messaoud et al. [2006] expand association rule mining from market basket data into a more multi-level data cube structure Chen et al. [2005]; Messaoud et al. [2006].

2.4.3 Rough Set Theory

Rough set theory was introduced by Pawlak [1991]. It is a mathematical approach that deals with the discernibility of objects and Boolean reasoning in information systems. Rough set theory (RST) or granule mining has the ability to discern differences and similarities between objects in most efficient way which can be useful for many KDD applications. RST has become one of the ten most popular theories applied by the data mining community. The successful contribution of RST in different applications including classification, association rules, dimensional reduction, patterns extraction and others, has demonstrated its importance and versatility. The fundamental concept of RST is described in Pawlak

[1991, 2002]. Paper Pawlak and Skowron [2007a] provides a survey analysis of all well known literature on data mining machine leaning and other knowledge communities that adopt RST.

The philosophy of RST depends on the assumption that with every object in the universe there is a certain level of indiscernibility among some information (data, knowledge). Using RST, users can describe the knowledge in the information tables Pawlak and Skowron [2007b]or multi-tier structures Li [2007]; Li and Zhong [2003]; Yang et al. [2008]. Additionally, users can represent the association among the data.

A transaction database can be formally described as an information table (T, V^T), where T is the set of transactions, and $V^T = \{a_1, a_2, \ldots, a_n\}$ is the set of items (or called attributes) for all transactions in T. For example, the following transactions: $t_1 = a_1a_2$; $t_2 = a_3a_4a_6$; $t_3 = a_3a_4a_5a_6$; $t_4 = a_3a_4a_5a_6$; $t_5 = a_1a_2a_6a_7$; and $t_6 = a_1a_2a_6a_7$ can be read as an information table (T, V^T), where $T = \{t_1, t_2, \ldots, t_6\}$ and $V^T = \{a_1, a_2, \ldots, a_7\}$.

Let X be an *itemset*, a subset of V^T. Its *coverset* is the set of all transactions (or objects) $t \in T$ such that $X \subseteq t$, and its support is $\frac{|coverset(X)|}{|T|}$. An itemset X is called *frequent pattern* if its support $\geq min\_sup$, a minimum support. Given a set of transactions (objects) Y, its *itemset* denotes the set of items (attributes) that appear in all the objects of Y. Given a pattern X, its closure $closure(X) = itemset(coverset(X))$.

Definition 1 A pattern X is *closed* if and only if $X = closure(X)$. X is called a max closed pattern if its all super patterns are non-closed, where pattern Y is called a super pattern of X if $Y \supset X$.

We now turn to discuss decision tables and granules. To easily understand

Table 2.1: A relational table

Object (*Transaction*)	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_1	1	1	0	0	0	0	0
t_2	0	0	1	1	0	1	0
t_3	0	1	0	1	0	1	1
t_4	0	1	0	1	0	1	1
t_5	1	1	0	0	0	1	1
t_6	1	1	0	0	0	1	1
t_7	0	1	0	1	0	1	1
t_8	1	1	0	0	0	1	1
t_9	0	0	1	1	0	1	0
t_{10}	0	0	1	1	1	1	0
t_{11}	1	2	2	0	1	0	2
t_{12}	2	1	1	2	2	2	0
t_{13}	0	2	1	2	2	2	2

the meaning of granules, we use the "Group By " operation with an example. We firstly represent the example information table (T, V^T) as a relational table (Table 2.1). We then use the following SQL statement:

select*, count(*) as *sup* group by $a_1, a_2, a_3, a_4, a_5, a_6, a_7$

to group the relational table into a decision table (Table 2.2) which classifies the seven attributes into condition attributes, $C = \{a_1, a_2, a_3, a_4, a_5\}$, including high spares attributes where the the number of granules in decision table is almost as the same on number of records in transaction table; and vice versa in decision attributes, $D = \{a_6, a_7\}$, which include less spares attributes.

Definition 2 Formally, we call the tuple $DT = (T, V^T, C, D)$ a decision table of (T, V^T) if $C \cap D = \emptyset$ and $C \cup D \subseteq V^T$.

Each row in the decision table is a granule, which can be viewed as a predicate that describes common features of a set of objects (transactions) for a selected set of attributes (or items). For example, in Table 2.2, the set of granules $U = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8\}$, where *sup* is the number of objects (or transactions) that have the same attribute values and *coverset* is the set of objects (or

Table 2.2: A decision table

G	*Set of Attributes*							*sup*	*coverset*
	Condition					*Decision*			
	a_1	a_2	a_3	a_4	a_5	a_6	a_7		
g_1	0	0	1	1	1	1	0	1	$\{t_{10}\}$
g_2	1	1	0	0	0	0	0	1	$\{t_1\}$
g_3	0	0	1	1	0	1	0	2	$\{t_2, t_9\}$
g_4	0	1	0	1	0	1	1	3	$\{t_3, t_4, t_7\}$
g_5	1	1	0	0	0	1	1	3	$\{t_5, t_6, t_8\}$
g_6	1	2	2	0	1	0	2	1	$\{t_{11}\}$
g_7	2	1	1	2	2	2	0	1	$\{t_{12}\}$
g_8	0	2	1	2	2	2	2	1	$\{t_{13}\}$

transactions) that are used to produce a group (or granule).

Simply, the set of granules G of the decision table is denoted by T/B, where $B = C \cup D$ is a subset of V^T. The granule in T/B that contains transaction t is denoted by $B(t)$. We describe the relationships between granules and transactions in formal concept analysis by using a relation R_B between $G = T/B$ and T. For a pair of transaction $t \in T$ and granule $g \in G$, if $g = B(t)$ (also written as tR_Bg), we say g is induced by t or t has the property g. The value of the granule g for attributes a_i denotes as $g(a_i)$

2.4.4 Rule Generation

Every granule in the decision table can be mapped into an *association rule* (or called *decision rule*), for example, the second granule, g_3, can be read as the following decision rule:

$$(a_1 = 0 \wedge a_2 = 0 \wedge a_3 = 1 \wedge a_4 = 1 \wedge a_5 = 0) \rightarrow (a_6 = 1 \wedge a_7 = 0)$$

where the antecedent and consequent are described as Boolean expressions.

Table 2.3: *C-Granules*

C-Granule	a_1	a_2	a_3	a_4	a_5	*sup*	*coverset*
cg_1	0	0	1	1	1	1	$\{t_{10}\}$
cg_2	0	0	1	1	0	2	$\{t_2, t_9\}$
cg_3	0	1	0	1	0	3	$\{t_3, t_4, t_7\}$
cg_4	1	1	0	0	0	4	$\{t_1, t_5, t_6, t_8\}$
cg_5	1	2	2	0	1	1	$\{t_{11}\}$
cg_6	2	1	1	2	2	1	$\{t_{12}\}$
cg_7	0	2	1	2	2	1	$\{t_{13}\}$

Table 2.4: *D-Granules*

D-Granule	a_6	a_7	*sup*	*coverset*
dg_1	0	0	1	$\{t_1\}$
dg_2	1	1	6	$\{t_3, t_4, t_5, t_6, t_7, t_8\}$
dg_3	1	0	3	$\{t_2, t_9, t_{10}\}$
dg_4	0	2	1	$\{t_1 1\}$
dg_5	2	0	1	$\{t_1 2\}$
dg_6	2	2	1	$\{t_1 3\}$

The smallest granules contain only a single attribute, we also call them primary granules. A large granule can be generated from some smaller granules by using logic "and", $\wedge$. For example, we can define small granules based on condition attributes and decision attributes using the following SQL statements:

select $a_1, ..., a_5$, count(*) as *sup* group by a_1, a_2, a_3, a_4, a_5

select a_6, a_7, count(*) as *sup* group by a_6, a_7

The former are *C-granules* (condition granules) and the *D-granules* (decision granules). Table 2.3 and Table 2.4 show the *C-granules* and *D-granules*, respectively; and the large granules $g_1, g_2, g_3, g_4, g_5, g_6, g_7$ and g_8 can be generated by *C-granules* and *D-granules*, for example, $g_1 = cg_1 \wedge dg_3$; $g_2 = cg_4 \wedge dg_1$; $g_3 = cg_2 \wedge dg_3$; $g_4 = cg_3 \wedge dg_2$; $g_5 = cg_4 \wedge dg_2$; $g_6 = cg_5 \wedge dg_4$; $g_7 = cg_6 \wedge dg_5$; and $g_8 = cg_7 \wedge dg_6$.

Definition 3 Let B be a subset of V^T and $G = T/B$, and granule $g \in G$ be

induced by transaction t. Its covering set $coverset(g) = \{t' | t' \in T, t' R_B g\}$.

It is then easy to prove the following theorem based on the above definitions:

Theorem 1 Let granule $g = cg \wedge dg$, where cg is a *C-granule* and dg is a *D-granule*. We have $coverset(g) = coverset(cg) \cap coverset(dg)$.

2.5 Data Quality in Context of Outlier Dimension

Outlier detection has become an important topic discussed in reference to many data mining and knowledge discovery applications, including those used for: credit card fraud detection, network intrusion, and virus detection. This section endeavours to investigate the limitations of the existing literature, specifically in the dimension of outlier data. The main goal of the thesis is to provide a complete automated approach for outlier data. Hence, the discussion centres around four areas that play an essential role in relation to automated data quality for outlier data. Outlier literature also considers other related approaches and phases of quality assessment and improvement because these approaches present general solutions that can be applied to any other type of poor data, including outlier.

2.5.1 Outlier Definition

Outlier definitions for applications involving data mining, credit card fraud detection, network intrusion, and virus detection can vary depending on the techniques used to define and detect outlier data. A broad definition of outlier data is data that has abnormal or malicious behaviour which is significantly different from the

remainder data. Specifically, outlier data is defined according to the technique used to detect outliers. The following points define outlier data based on the techniques used:

- Statistical-Based: The three-sigma rule for normal distribution of data is the method used to define outlier points from normal data. Based on the three-sigma rule, outliers are points that have more than 3 standard deviations Hawkins [1980].

- Distance-Based: In a distance-based approach, the outlier is "an object O in dataset T is a DB(p, D)-outlier if at least fraction p of the objects in T lies greater than distance D from O " Knorr and Ng [1998, 1999].

- Density-Based: The outlier points in density based methods are based on the local density of an objects' neighbourhood. Formally, the outlier point p in database T is the ratio of its density to the average density of its neighbours Breunig et al. [2000].

- Clustering: Clustering technique defines an outlier cluster as a small cluster that is far away from the other major clusters Xiong et al. [2006]. Within the cluster, the outlier instances are the instances that are furthest distant from their corresponding cluster centroids.

- Classification: Outlier or (noise) examples are instances that are mislabelled in the training datasets [Brodley et al., 1996].

- Pattern Mining: The outlier pattern is defined as a pattern with item sets that infrequently occur in a data set He et al. [2005]; Otey et al. [2006].

2.5.2 Outlier Detection

Several domains of knowledge have investigated outlier problems and proposed a number of interesting solutions. Historically, most outlier detection approaches originate in the statistical domain, which is also called statistical-based or model-based outlier detection. Most statistical-based methods are univariate outlier detection methods, which detect outlier values using a single attribute or feature. Additionally, a statistical-basis requires a prior knowledge of the data distribution Barnett [1978]; Hawkins [1980]. These limitations make a statistical-based approach that is impractical for detecting outliers in most KDD applications. The reason for this is associated with the fact that underlying data distribution is usually unknown and expensive to compute in most KDD applications. Also determining the distribution of these KDD applications is computationally expensive because of the data size and dimensionalities of KDD applications. Other statistical studies following a depth-based approach do not require prior knowledge of the data distribution Aggarwal and Yu [2001]; Johnson et al. [1998]; Ruts and Rousseeuw [1996]. However, the depth-based approach is only useful for $K < 5$ dimensions (or attributes) and is inefficient for more dimensional data sets because of the computing required for construction of the appropriate convex hulls.

An additional dimension of outlier research is that which focuses on distance-based outlier detection. In this case, the outlier is defined as "an object O in dataset T is a DB(p,D)-outlier if at least fraction p of the objects in T lies grater than distance D from O " Knorr and Ng [1998, 1999]. The authors present three different distance-based algorithms: index-based, cell-based and partition-

based. Unlike the statistical-based method, the distance-based method does not require any a prior knowledge of data distribution. Furthermore, distance-based scales well with the increase in k dimensions. However the major drawback of these methods relies on the impact on distance-based of complexity related to time. The running time for a nested-loop in index-based algorithms, as used in a distance-based approach is quadratic in the input size $O(KN^2)$. Authors Knorr and Ng [1998, 1999] improve the time by using a cell-based algorithm that is linear with respect to the size of the database, but the cell-based algorithm is exponential with a dimension $k > 4$. Additionally, it is difficult to correctly specify the distance parameters. An interesting distance-based method is presented in Ramaswamy et al. [2000]. The authors use the distance of the kth nearest neighbours to rank outliers based on outlier degrees, instead of binary labels used in Knorr and Ng [1998, 1999]. Another distance-based method sums the weight of each point from its nearest neighbours and used it to rank outlier objects Angiulli and Pizzuti [2002]. The state-of-art in the distance based method for outlier detection was introduced in Bay and Schwabacher [2003]. The study described in Bay and Schwabacher [2003] presents an efficient distance-based algorithm called Orca with near linear time performance. Unlike other distance-based methods, the Orca algorithm Bay and Schwabacher [2003] mines numeric and mixed attribute data types. Another efficient method of distance-based methods is presented in Ghoting et al. [2008]. Some methods have attempted to improve the performance of the distance-based approach by using index structure R*tree Beckmann et al. [1990], as a partition with a clustering algorithm McCallum et al. [2000]; Ramaswamy et al. [2000]. However, both index and partition clustering do not scale well with increases in database dimensions.

Density-based outlier detection is another direction for outlier research. This method was presented in Breunig et al. [2000]. The authors introduced the notation for the local outlier factor (LOF), which captured the degree of outlier-ness for each object. The outlier point p in database T is comprised of the ratio of its density to the average density of its neighbours Breunig et al. [2000]. However, the main drawback of this method is the sensitivity involved in specifying the number of nearest neighbours *MinPts* as a parameter. A recent study in Papadimitriou et al. [2003]] attempts to address the problem of *Minpts* by introducing probabilistic reasoning. Paper Jin et al. [2006] overcomes the problem of different density distribution in the neighbours from Breunig et al. [2000] and presents a simple measure for local outliers based on a symmetric neighbourhood relationship. Other studies that investigate the problem of the density of the neighbourhood are presented in Jin et al. [2001, 2006]; Liu et al. [2010]; Tang et al. [2002].

Other far-reaching work on outlier detection depends extensively on clustering algorithms. In the case of a clustering algorithm, any point that does not belong or fit into the cluster is labelled as outlier or (noise in the data mining community). However, the accuracy of the clustering algorithms depends on the parameters applied, e.g. number of clusters. Additionally, there is no degree of outlier-ness that is applied to define objects as outliers property in clustering algorithms is with binary. Some approaches use clustering algorithms with a distance-based method McCallum et al. [2000]; Ramaswamy et al. [2000] to enhance performance. However, their solutions have serious limitations regarding increased dataset dimensions. Density-based outlier methods depend to some extent on clustering capabilities; but drawbacks arise when specifying *Minpts*

nearest neighbours as parameters, as well as the difference in cluster distribution.

An interesting approach to outlier detection is based on finding frequent itemsets such as those introduced in Agrawal and Srikant [1994]. The algorithm Agrawal and Srikant [1994] makes multiple passes over the data in order to discover frequent itemsets. This step results in the generation of a large number of itemsets. The situation becomes much more complicated when dealing with large quantities of dimensional data when the support threshold is very low. Hence, this step will become computationally expensive with regards to performance and memory usage.

Recently, there has been growing attention directed towards utilising Frequent Itemsets Mining (FIM) for outlier detection. The outlier in FIs is defined as the patterns that infrequently occur in data He et al. [2005]; Otey et al. [2006]. Both algorithms FPOF and Otey He et al. [2005]; Otey et al. [2006] assign outlier degrees for each point. An updated algorithm of He et al. [2005], based on a fast greedy algorithm for outlier detection was presented in He et al. [2006]. Yet, these algorithms He et al. [2005]; Otey et al. [2006] depend on Agrawal and Srikant [1994] for finding all frequent itemsets. Furthermore, users need to carefully specify a minimum support threshold. Although the greedy algorithm presented in He et al. [2006] is fast, users have to initially determine the desirable number of outliers that the greedy algorithm should return. Paper Koufakou et al. [2011] reduces the numbers on an itemset by using Non-Derivable Itemsets and an approximation of Non-Derivable Itemsets techniques. However, the accuracy of outlier detection is dependent on the specified minimum support threshold. A study in paper Koufakou et al. [2007] presents Attribute Value Frequency AVF from the transaction dataset for categorical data types. In the AVF algorithm,

the AVF scores for x_i is the sum of all attributes' frequencies in x_i. The strongest outlier point is the point that has lowest total of all AVF attributes. The AVF algorithm scales compare well to the previous frequent itemsets as AVF does not compute itemsets. However, the use of the AVF algorithm, without considering the frequency of patterns can provide incorrect outlier detection and ranking. For example, if a user has two points x_i and x_j with equal AVF scores, then the AVF algorithm cannot determine which one is the strongest outlier. Further, the AVF may incorrectly flag the outlier point because the point that ranks top might appear more frequently than other points. Hence, it is essential to consider frequent patterns with item frequency for correct outlier detection. Most of the frequent itemsets methods are suitable for use with categorical data types, except for the few algorithms presented in Koufakou et al. [2008]; Otey et al. [2006]; Yu et al. [2006] which detect outlier data from mixed attribute datasets.

Despite the popularity of RST in data mining and machine learning, it represents only a small contribution to studies, in terms of those adopting it for outlier detection. In paper Jiang et al. [2005], the authors exploit the framework of RST for outlier detection. The extended version of their method was presented in Jiang et al. [2009]. Their approach Jiang et al. [2009] presents sequence-based outlier detection in accordance with RST. The authors assume that there is a function between attributes. Their study partitions the set of attributes into two parts: Condition attributes and Decision attributes. A similar study using granule mining is paper Chen et al. [2008]. The authors of this research also assume that there is a function between the attributes from the information table which are also referred to as a decision table. The use of the RST or granule miming finds item sets more efficiently than would be possible computing all frequent

items as Koufakou et al. [2008]; Otey et al. [2006].

2.5.3 Quality Assessment

2.5.3.1 Quality Assessment Phases

Quality assessment is an essential step after defining and detecting outlier data. It enables decision makers and management to track down the areas that produce quality conflicts. Quality assessment enables users to anticipate what and where efforts should be made to improve data quality. Redman states, that "which does not get measured does not get managed " Redman [1998]. The awareness of the quality level of a database or a data warehouse enables management to capture the root causes of quality problems in general and of outlier data in particular. Moreover, it reduces the impact of outlier data and creates a roadmap for achieving improvements to quality.

Data quality in a database or data warehouse has to be assessed and monitored continuously to ensure a high level of quality is maintained Cappiello et al. [2005]. Proper quality assessment for outlier data can benefit users by assisting this monitoring by reflecting the change of data quality status with regards to outlier data periodically. Users can then determine whether or not there is degradation or improvement in data quality and therefore determine the usefulness of existing techniques for improvement. Additionally, quality assessment is significantly beneficial in situations where managers cannot or do not have the resources to improve data quality, since quality assessment can estimate the costs or risk of the decisions made based on incorrect data quality.

To address this issue, the literature provides both data-oriented and process

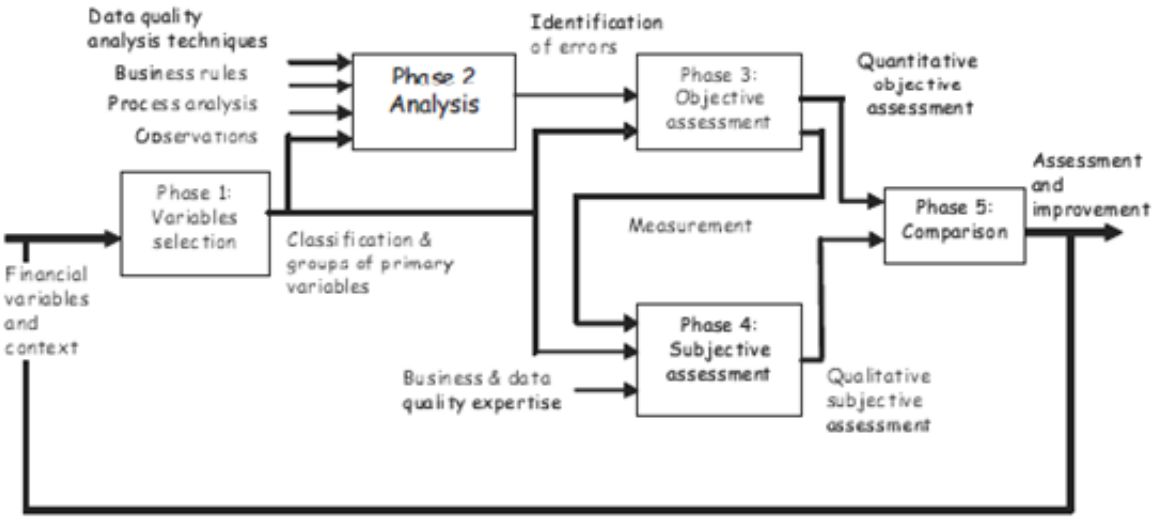

Figure 2.6: The main phases of the assessment methodology Batini and Scannapieco [2006]

oriented techniques. Batini Batini and Scannapieco [2006] classifies data quality assessment into five phases including: variables selection, analysis, objective assessment, subjective assessment and comparison.

- Phase 1: variables selection, concerns the identification, description and classification of primary variables that are most relevant to databases. Then, these are characterised, according to their meaning and role into qualitative/categorical, quantitative/numerical, and date/time.

- Phase 2, analysis of data dimensions using some statistical techniques to inspect the integrity of the data. The results of this analysis lead to presenting deficient data with regards to quality dimensions.

- Phase 3, objective assessment, presents the quality level of information

systems by calculating the ratio of erroneous observations for different dimensions.

- Phase 4 deals with subjective assessment. Qualitative assessment is based on the consumers who ultimately decided if the data values are correct or not. Subjective assessment is usually obtained by merging independent evaluations from stockholders including a business expert, who analyses data from a business process point of view, and data quality experts, who take on the role of analysing data and examining its quality.

- Finally, a comparison between objective and subjective assessment is performed to determine that both quality assessment attain the required level of consent necessary to nominate data values as poor quality.

These two views of quality assessment have some advantages and limitations that are discussed in detail in the next section.

2.5.3.2 Quality Assessment Methods

It is well accepted in the data quality literature that assessment of the status of the quality of information systems cannot be achieved independently from the perspectives of the data consumers (or users). Several outstanding theoretical methodologies have been proposed to investigate quality assessment and improvement from the users perspective. These methods are primarily reliant on qualitative assessment methods such as questionnaires, surveys, and interviews Lee et al. [2002]; Wang and Strong [1996]; Wang and Kon [1993]. In this way the results of subjective assessment will be obtained from information collectors, information consumers, and IS professionals.

However, quality assessment based on a user perspective is subjective according to the users behaviour. A study in a major U.S. bank based on subjective assessment (questionnaire) revealed that different assessment results of data quality can be obtained from custodians (IS professionals) who view their data as clean and high quality and from consumers who view data as difficult to utilise for business purposes Huang et al. [1999]. Subjective assessment can then introduce inaccurate assessment of quality dimensions Pipino et al. [2002]. Assessing data based on a users perspective is a difficult and time consuming task. It requires manual inspection by the users of data values to determine which ones are of poor or high quality. Furthermore, such subjective assessment is an unsuitable solution for ongoing data quality monitoring in a database or data warehouse.

An alternative method of assessing data quality from the user perspective is data perspective. Utilising the data perspective for quality assessment has a great advantage when providing objective quality assessment, automated approach, and time and cost efficient as the assessment derived from the data. For objective quality assessment, the data values must be converted to binary format (0 representing normal data and 1 defective data (i.e. outlier data in this study)). Most existing quality assessment methods transfer their data matrix to binary matrix. This study also applies the same procedure and converts our dataset to 0 for normal value and 1 for outlier values after exposing outlier values based on the proposed algorithms.

After the data has been converted to a binary matrix, the user can conduct quality assessment tasks. Most common objective quality assessment methods rely on error rate to calculate the ratio between the number of correct values and the total number of values for targeted databases or data warehouses. After

deficient data values have been disclosed, users can easily employ the error rate method by counting the total number of defective data fields and then dividing this by the total number of fields Batini and Scannapieco [2006]; Dasu and Johnson [2003]; Pipino et al. [2002]; Redman [1996]. Users can also calculate the accuracy rate by subtracting 1 from the result of the error rate "accuracy rating = 1- (total numbers of defective data/total numbers of fields)". This method is useful for presenting the ratio of normal or abnormal data values.

However, an error rate or accuracy rate assessment approach is an insufficient and inefficient objective assessment. This method offers no guidance information to managers to assist them in improving data quality. Further, the error rate method might report the same error or accuracy ratio as that in the databases. This result might be misleading since errors are randomly and systematically distributed across databases. For instance, if management wants to assess the quality of three databases or the quality of data over the last three years, the error rate method might report identical results. This might be incorrect because errors rates may be for the first database systematically distributed in one column and systematically distributed in one row for the second database and randomly across columns and rows for the third database.

Paper Pipino et al. [2002] presents both a subjective and an objective approach to quality assessment. The view has been experimentally extended Even and Shankaranarayanan [2009] and presents objective assessment as impartial and subjective assessment as contextual. In contextual assessment users assess quality based on the context or the intention behind the analysis of the data. However, this method evaluates data quality in a single tabular data set and is not suitable for a multi relational database or a data warehouse; hence, detecting defective

values and quantifying their impact can be challenging.

An error rate approach does not provide managers with valuable information; such as quality change from the previous year to the current year. Managers and executives cannot rely on the error rate method to conduct or estimate quality improvement time and costs. Additionally, managers cannot apply the error rate method when comparing quality issues across databases because errors are randomly and systematically distributed across columns and rows. For instance, if management wants to assess the quality of three databases A, B, and C or the quality of data over the previous three years, the error rate method might report the same error rate results. This might be misleading since errors can be distributed in different locations.

For example, if the error rate is 5% for all databases A, B, and C. The time and the cost needed to correct these errors varies because errors in databases A and B are systematically distributed across columns and rows respectively. Whereas, errors in database C are randomly distributed across columns and rows. Therefore, it is essential to measure the systematic and random degree of error data rather than only focusing on presenting the rate of defective data.

Another well know study that has resulted in a major breakthrough in data quality assessment, particularly for missing data was presented in Little and Rubin [2002]; Rubin [1976]. The interesting thing in this study, rather than the methods described for handling missing data, which is out of our scope in this thesis, is that the authors paid some attention to quality assessment. Specifically, in Little and Rubin [2002], the authors distinguish the missing data patterns which describe the location of missing data and missing data mechanisms, which concern the relationship between the absent data and the values of variables in

the data matrix. Their study classifies the missing data mechanisms into three types, missing complete at random (MCAR), missing at random (MAR) and not missing at random (NMAR).

Formally, paper Little and Rubin [2002] defines the complete data as $Y=(y_{ij})$ and the missing data as $M=(m_{ij})$. The missing data mechanism is classified by the conditional distribution of M given Y, say $f(M|Y, \phi)$ where ϕ denotes unknown parameters. If the being missing does not depend on the values of Y, then the missing data is characterised as MCAR. Another missing data mechanism is MAR, which is less restrictive than MCAR as the missingness depends only on the observed data. The last one is NMAR where the missingness depends on the missing attribute. We refer the reader to Little and Rubin [2002] and other supplementary resources for more details about missing data mechanisms Allison [2001]; Amanda and Craig [2010]; Enders [2010]; Schafer and Graham [2002]. These classifications are useful for determining which handling methods; such as maximum likelihood, multiple imputation and other methods are most appropriate to predict missing values. However, although, this solution Little and Rubin [2002] is good for dealing with small data sets with few attributes, it is inefficient for large dimensional database because it generates too many cases. Additionally, in practice, it is hard and implicit to only use probability to confirm that missing data solely functions according to other attributes. Association rules discussed in this paper presents more precise and clearer solutions for ascertaining the relationship between attributes and missing attributes than probability.

A recent study Fisher et al. [2009]proposes a new quality assessment method. The authors extend the current error rate method to include a randomness measure and probability distribution to enhance the quality assessment from a data

perspective. In their study, quality measurement is based on three vectors; including error rate, randomness measures and probability distribution. Firstly, they calculate error rate or accuracy rate based on the previous definition in the previous paragraph. Then, the authors adopt a Lempel-Ziv (LZ) complexity algorithm in order to state whether the errors in the databases are randomly or systematically distributed. Finally, the study adopts the Poisson distribution method to measure the probability of errors in a database, due to the fact that some errors are greater in some rows and columns than others.

However, this study is inefficient for the assessment of a large database. The problem, as the authors Fisher et al. [2009] indicate, is that the Lempel-Ziv (LZ) algorithm has a complexity associated with time that makes it inadequate when assessing a large database. Therefore, it is suggested that a database should be randomly segmented into two small samples to compute the LZ algorithm. In this case users have to run the LZ algorithm several times, which is impractical and inefficient if dealing with a large database or a data warehouse. More importantly, this method does not allocate the location of defective data. Users do not exactly know where the problems occur or how severe their degree, in terms of impact on the database or data warehouse.

2.5.4 Quality Improvement

Previous sections delved into procedures for defining, detecting and assessing outlier data. This section investigates the important of designing a data cleaning solution to the dimension of outlier data. Users can utilises outlier detection methods to distinguish between normal data and exceptions (or outliers). Based

on the results the user can, efficiently and effectively design a model to trigger indication of suspicious data. Or the model can find an association between values that indicates any suspicious behaviour. These benefits could be important when designing a data cleaning solution to clean and maintain high data quality in a database. However, the challenge is how to effectively and efficiently capture the outliers so that the user can build a model that improves the quality of tasks.

Similar to quality assessment, quality improvement can be classified into two types: process-driven and data driven. Typical process-driven models measure the current data status, identifying existing integrity constraints, dirty records, and then developing new constraints Fei and Ren [2008]. To perform these tasks, consultation with domain experts and users who have specific knowledge of business rules must be established. This cannot be easily achieved with high accuracy and minimum cost for two reasons. Firstly, consulting with people requires an extensive amount of time and effort which consequently increases the cost of conducting data quality improvement tasks. Secondly, there may some existing rules in the data that users are not aware of which may lead to Inconsistent data Fei and Ren [2008].

A data-driven approach focuses on the data itself triggering the outliers and suspicious behaviour from accessing the database and updating the noise of the data in order to improve the accuracy of the data analysis task.

2.5.4.1 Quality Improvements Based on Rules Techniques

Rule discovery based on an existing relationship is an important technique used in database design and data mining. It helps to discover the dependency between attributes in a database relationship Fan et al. [2008]. Originally, this technique

was motivated by the fact that data mining concerns finding interesting patterns in large data sets. Additionally, rule discovery was motivated by expressing constraints on a database or schema level Wyss et al. [2001]. Recently, rule discovery has been extended further to address data quality problems Fei and Ren [2008]. By applying rule discovery, violated rules can then be uniquely determination based. Therefore, this technique is essential not only for identifying dirty and inconsistent data values and suggesting possible rules to replace the abnormal values, but also for accelerating the data cleaning process.

Unlike other data quality algorithms, rule discovery can efficiently expose unknown data quality rules to domain knowledge experts. This easily enables these experts to correctly determine whether or not a nominated record has incorrect values (e.g. a 'husband' who is a 'female'). The idea that underpins this approach; i.e. discovery of normal or abnormal data relies on the fact that normal or abnormal data values can be determined by considering the relationship and dependency between attributes values Wenfei et al. [2009]. However further research is needed to implement rule discovery in the data cleaning process, and in particular to specify integrity constrains to model the semantics of the data level Rahm and Do [2000].

- **Functional Dependencies**

Out of the discovery rule dependencies emerge; one of these referred to as functional dependencies (FD). Functional dependencies define the relationship between the attributes of a database where the value of an attribute is uniquely predictable based on other attributes' values Bitton et al. [1989]; Huhtala et al. [1999]; Mannila and Raiha [1992]. Functional dependency (FD) includes discov-

ery of functional, multi-valued and approximate data. A formal definition of functional dependency (FD) over a relation schema R is an expression $x \rightarrow y$ where X and Y be subsets of a relational schema R. A functional dependency (FD) $X \rightarrow Y$ asserts that any two tuples that agree with the values of all the attributes in X (the antecedent) must agree with the values of all the attributes in Y (the consequent) Golab et al. [2008].

The best and most well known method of FD Discovery for functional use is TANE Huhtala et al. [1998]; this is followed by other methods: DepMiner Lopes et al. [2000] and FASTFDs Wyss et al. [2001]. TANE is based on partitioning a set of rows with respect to their attribute values. This helps to test the validation of the functional dependencies quickly regardless of the large number of tuples. TANE, and subsequent approaches such as DepMiner are passed on the breadth-first or levelwise algorithm for searching the attributes lattice for minimal FD cover Huhtala et al. [1999]; Lopes et al. [2000]. In these studies the authors demonstrate how a search space can be pruned and how to compute partitions and functional dependence (FD).

Similarly to TANE , DepMiner Lopes et al. [2000] adopts the same levelwise algorithm; the only difference being that DepMiner reduces the FD discovery problem when finding minimal covers for hypergraphs and then applies a levelwise algorithm. Another study presented by Wyss et al. [2001] introduces the FASTFD algorithm, where the authors employ a depth-first-algorithm for searching attributes. The experimental study in Wyss et al. [2001] proves that a depth-first algorithm surpasses a levelwise searching strategy, but it is more sensitive to the size of the relations and data complexity. Paper Savnik and Flach [2000] studies functional dependency (FD) in cases of multi-valued dependencies from

relations. The authors present two algorithms, a top-down and a bottom-up one for the discovery of multi-valued dependencies from relations Savnik and Flach [2000]. However, the reason that the TANE method is the best is that it solves two fundamental concerns, discovery of functional dependency and approximate functional dependencies.

In paper Huhtala et al. [1999], the authors also present an approximate FD which is useful to find dependency between attributes that contain errors or represent exceptions to the rules. FD and approximate FD are useful to predict and correct defective data. For example gender values can be determined by a first name. Similarly, zip codes can be determined by the values of country, city and street name attributes. However, FD and approximate FD approaches are unable to adequately improve data quality at a data level. These approaches, which depend on traditional dependencies (e.g. functional and full dependencies, etc) were mainly developed for a schema design and are often insufficient to capture semantic problems at the data level Fei and Ren [2008].

- **Conditional Functional Dependencies**

Conditional functional dependency (CFD)is an extend method of functional dependency (FD). Conditional functional dependencies (CFDs) have recently been proposed as a useful integrity constraint to summarise data semantics and identify data inconsistencies Golab et al. [2008]. Conditional functional dependencies (CFDs) enables users to improve the data quality at both the intensional (schema) level and extensional (data) level and also to capture any suspicious values that breach CFD rules Bohannon et al. [2007]. Additionally, CFDs hold conditionally, that is, only on the subset of a relation that satisfies certain pat-

terns, rather than on the entire relation.

Recently algorithms have been presented in Fei and Ren [2008]; Golab et al. [2008] for the purpose of discovering CFDs. In paper Golab et al. [2008], the authors use support, confidence and parsimony to present the discovery of good patterns. They define and study the complexity of discovering optimal patterns in a tableau generation. Paper Fei and Ren [2008]proposes an algorithm that aim to discover approximate conditional functional dependency rules and then identifies exceptions to these rules. The proposed algorithm searches for minimal CFDs among the data values and prunes redundant candidates.

Consider the following example in Figure 2.7 where the cust relation consists of the these attributes (CC, AC, PN, NM, STR, CT, ZIP), (country code (CC), area code (AC), phone number (PN)), name (NM), and address (street (STR), city (CT), zip code (ZIP)). Traditional functional dependencies (FDs) on a cust relation may include:

f1: $[CC, AC, PN] \rightarrow [STR, CT, ZIP]$

f2: $[CC, AC] \rightarrow [CT]$

f1 requires that customer records with the same CC, AC and PN also have the same STR, CT and ZIP. Similarly, f2 requires that customer records with same CC and AC also have the same CT. Traditional FDs are to hold on all the tuples in the relation.

Unlike FD, constraints ϕ_0,ϕ_1,ϕ_2, and $\phi_3$are FD$_s$ that is hold on to the subset of tuples that satisfy the pattern rather than on the entire cust relation, as can be seen in the following: These enables user to capture a fundamental part of the semantics of the data.

(a) Tableau T1 of $\varphi 1$ =(cust[CC,ZIP] → [STR], T1)

CC	ZIP	STR
44	-	-

(b) Tableau T2 of $\varphi 2$ =([CC, AC,PN] → [STR, CT,ZIP], T2)

CC	AC	PN	STR	CT	ZIP
-	-	-	-	-	-
01	908	-	-	MH	-
01	212	-	-	NYC	-

(c) Tableau T3 of $\varphi 3$ =([CC,AC] → [CT], T3)

CC	AC	CT
-	-	-
01	215	PHI
44	141	GLA

Figure 2.7: CFDs Example Fei and Ren [2008]

$\phi_0 : [CC = 44, ZIP] \rightarrow [STR]$

$\phi_1 : [CC = 01, AC = 908, PN] \rightarrow [STR, CT = MH, ZIP]$

$\phi_2 : [CC = 01, AC = 212, PN] \rightarrow [STR, CT = NYC, ZIP]$

$\phi_3 : [CC = 01, AC = 215] \rightarrow [CT = PHI]$

The first ϕ_0 requires that customer records with the same CC pattern (44) and the same ZIP code have an STR name. By observing the standard FD (f1 and f2) constraints ϕ_0,ϕ_1,ϕ_2, and ϕ_3,we find that $\phi_1 and \phi_2$ refine the standard FD f1 where as ϕ_3, refines the FD f2. These enhancements are essential to enforce bindings of semantically related data values Fei and Ren [2008]. Indeed, while tuples t1 and t2 in Figure 1 do not violate f1, they violate its refinement ϕ_1 , since the city cannot be NYC if the area code is 908.

These constraints ϕ_0,ϕ_1,ϕ_2, ϕ_3,f1 and f2 can be uniformly expressed in CFDs. By adopting CFD and representing both the data and the constraint in tableau format, as in Figure 2.7,the user will achieve two valuable goals, at one end of the spectrum are relational tables which are composed of data values without logic variables, and at the other end are traditional constraints which are defined in terms of logic variables but without data values, while CFDs are in the space in between Bohannon et al. [2007]. In Figure 2.7 , φ_1(for ϕ_0), φ_2 (for f1, ϕ_1 and ϕ_2, one per line, respectively) and φ_3 (for f2, ϕ_3 and an additional $[CC = 44, AC = 141] \rightarrow [CT = GLA]$.

However, the CFDs depend on a traditional Apriori algorithm Agrawal et al. [1993] to find the FD and CFDs associations. Finding the CFDs from the Apriori algorithm generate a large number of rules and conditions. This reduces the efficiency of the data cleaning solutions when triggering or updating outlier (noise) data.

2.6 Chapter Summary

This chapter has covered the literature that relates to data quality and the occurrence and management of outliers. It has investigated the limitations of existing data quality methods in reference to outlier data according to four areas: Data quality, knowledge discovery, data mining and outlier data. Considering these four areas in the literature review provides a comprehensive coverage of the research problems, existing methods and promising directions towards an automated data quality solution. From the literature, it can be clearly observed that data quality raises multi dimensional problems and that it has been investigated

by many applications. This results involve classifying data quality literature into two views: Process-oriented and Data-oriented. Each of these views has many advantages and limitations. However, there is a gap in terms of contribution towards the data-oriented view; particularly for outlier data. The problem with most data quality research based on data-oriented view is that data-oriented methods are limited to exposure and correcting poor data including outliers. This undoubtedly is a critical step towards high data quality. Yet, errors are going to continue to appear as these method fail to capture the root causes of the problems. Additionally, there are efficient and effective problems using traditional data mining techniques (such as classification or clustering) for outlier detection; such methods are mainly designed for frequent data.

The following chapter will provide a framework outlining the proposed automated data quality solution. The framework divides the problem of the existing data quality in outlier data into sub-problems which enable readers to follow the structure of the thesis's contributions when approaching these problems.

Chapter 3

A Framework of the Proposed Data Quality Solution

The purpose of this chapter is to introduce the framework for the proposed data quality solution for outlier data. The framework clearly depicts the roadmap of the thesiss contributions. As can be seen in Figure 3.1, there are three major contributions made by this research. These contributions are both gradually and continuously contingent on each other. Each of these three contributions deals with specific quality problems. The First division in Figure 3.1 concerns finding and extracting patterns based on RST; using it to facilitate an efficient and effective design for mining outlier data and quality assessment. The second part of the framework 3.1 shows three different algorithms that are designed to solve different problems that arise with outlier detection methods. The last division in the framework, section 3.1 is devoted to the assessment of outlier data. This part proposes a new way with which to assess data quality as well as to measure a systematic and random degree of outlier data. The contributions specified ac-

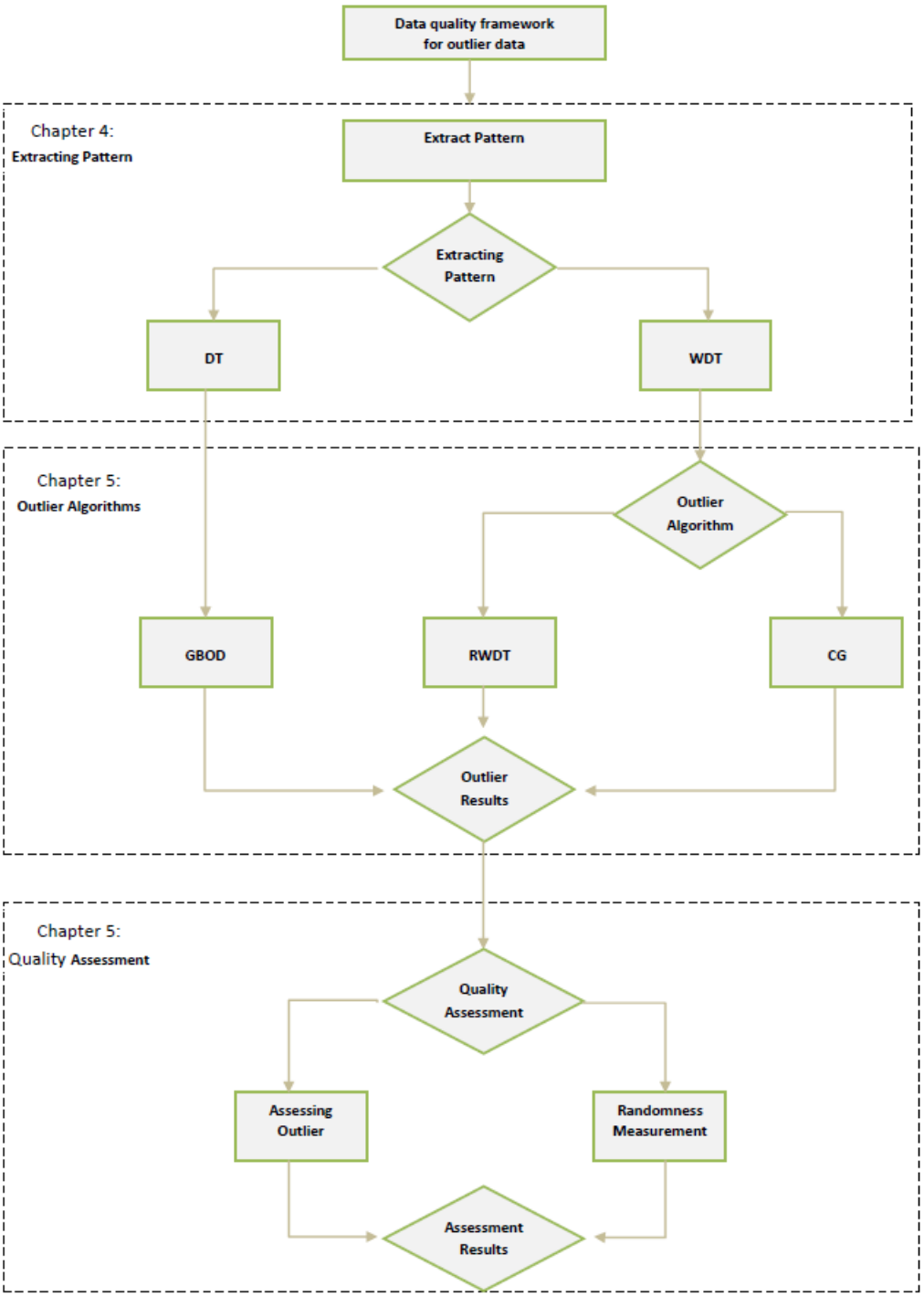

Figure 3.1: Data Quality Framework for Outlier Data

cording to these divisions each solve serious problems that arise with data quality. The framework of these contributions allows a breakthrough towards achieving a complete automated data quality solution for outlier data. The following will introduce those divisions shown in Figure 3.1 in further details.

3.1 Extracting Useful Patterns

This study utilises RST to extract a useful pattern based on evidence collected by reviewing literature on this topic. As detailed in Chapter 2 RST provides a novel approach to pattern extraction, and yet, in spite of this and its popularity, the theory has been rarely tested in reference to outlier data and data quality problems. The literature review in Chapter 2 highlights that the major problem that arises in traditional data mining tasks is the large number of patterns that emerge. This usually reduces the efficiency of those algorithms that are used to mine outlier data. In RST, the number of extracted patterns is lower and more useful, as there is less noise than with frequent pattern mining. To design a data quality solution, the thesis presents a traditional decision table DT and a novel way of extracting patterns, called a Weighted decision table WDT.

3.1.1 Decision Table (DT)

The use of a decision table, as will be discussed in Chapter 4, provides users with a clear insight into the density of objects in a dataset, because all objects are grouped in granules in a decision table DT with support sup. Based on the support granules the user can distinguish between granules (patterns) with possible outlier data, from patterns that are unlikely to hold outlier values or

which will assess the quality of changes in outlier data or measure systematic and random distributions of outlier data. Since the DT method results in a large number of granules with the same degree of support, defining outlier values in an efficient way is difficult.

3.1.2 Weighted Decision Table (WDT)

Distinct from the DT, the WDT method can be used to compute the weight of each item set that shared the same attributes and values, rather than focusing on computing the items. This new approach provides more meaningful information than the existing DT would, because it considers the weight of the each value in the information table. The use of the WDT method is critical for improving the effectiveness and efficiency of the $RWDT$ and CG outlier algorithms that will also be proposed in this work.

3.2 Algorithms for Outlier Detection

The benefits of identifying useful patterns and extracted them using both DT and WDT, have brought significant advantages with regards to outlier detection. This is because of the fact that patterns based on DT and WDT are much faster when compared to Apriori algorithm; the Apriori computes all item sets to find frequent pattern mining. Another reason for this is that the number of extracted patterns, found by DT and WDT is much smaller than the number of patterns found using the Apriori algorithm. These advantages contribute to the designing of three effective outlier algorithms.

3.2.1 Granule Based Outlier Detection (GBOD)

As can be seen from the framework given in Figure 3.1, the $GBOD$ algorithm is derived from the pattern found using DT. This algorithm proves that the discernibility matrix can provide accurate distance measurement for outlier detection, as accurate as the traditional Euclidean distance, if the discernibility matrix includes the weight when computing distances between patterns or points.

3.2.2 Ranking Weighted decision Table (RWDT)

The previous $GBOD$ algorithm provides an effective solution for the detection of outlier data using the new weighted discernibility matrix. However, since the number of approximate patterns in the DT is large, the $GBOD$ algorithm encounters efficiency problems with an increasingly larger dataset. To address this problem, the proposed thesis introduces $RWDT$ for outlier detection. The new algorithm utilises the WDT method to extract very small useful patterns. Mining such small patterns can improve the efficiency when mining outlier data in a large dimensional data set.

3.2.3 Centroid Granule (CG)

Most popular outlier methods compute the distance between a point to a set of nearest neighbouring points, in order to determine the similarity or deviations, degree and therefore determine outlier degrees. Also users need to carefully specify a number of parameters, such as minimum distance and number of neighbours. This is an efficient method by which to increase data size and dimensionalities. The new CG granules algorithm solves these issues by finding the centroid gran-

ules; then the distance is computed from approximate possible outlier patterns found by WDT and CG to assign outlier degree patterns.

3.3 Quality Assessment

The existing quality assessment fails to propose a reliable solution that assesses quality change and allocates the locations of the most severe data. The reason for this is that most assessment methods adopt error or accuracy rate methods. Although, such methods can usefully show the ratio of outlier data in a database, they might provide inaccurate assessment as errors can have different distributions. To overcome this problem, the framework in Figure 3.1 involves two methods: Decision Rule Method for Data Quality Assessment and the Randomness Degree.

3.3.1 Decision Rule Method for Data Quality Assessment

The proposed decision rule method provides management with information needed for data quality assessment. The proposed methods will determine any quality changes in or across different databases or data associated with different time periods. This will benefit users by determining the degree of improvement or any degradation in the quality of the data.

3.3.2 Randomness Degree

Since, poor data can appear in different locations in the database to different degrees, it is essential for an advanced data quality solution to measure distribution of these errors. This study designs an algorithm that highlights the most severe

outlier problems in the database by measuring the systematic and random degree of the outlier data. Accessibility to this type of measurement enables users to allocate the right resources to conduct quality improvement tasks.

3.4 Chapter Summary

This chapter detailed the framework for the proposed data quality for outlier data. The goal of the framework is to clearly illustrate the challenges associated with data quality research. The framework divides the problem by designing an automated data quality approach to outlier data in three parts. The first of these concerns solving the limitation of extracting useful patterns for resolving data quality problems. The second derives three different outlier algorithms. The final part presents two methods, one for data quality assessment and another for degree of randomness.

Chapter 4

Extracting Candidate Patterns

4.1 Introduction

Outlier detection is an essential requirement for many knowledge domains, including applications involving data mining, fraud detection, network intrusion. It plays a major role in improving the accuracy of data mining tasks, so as to prevent malicious and suspicious activities in both the public and private sectors and also to protect our privacy from breach by detection according to network intrusion. Despite these advantages of outlier detection applications, designing an effective outlier solution is an immense challenge. This is essentially because outlier data with significant deviation from the reminder data only infrequently appears in a database. Mining such data to expose infrequent data is computationally expensive.

Some studies have adopted data mining tasks to improve effectiveness and efficiency when mining outlier data. The literature in this area includes various interesting algorithms and techniques to deal with outliers and noisy data. How-

ever, there is a problem when deploying classic data mining solutions; including classification, clustering, and association rules for outlier detection. Since, these data mining tasks are primarily designed to deal with frequent or what interesting patterns. In data mining, extracting the knowledge from frequent patterns to answer users needs is a critical problem, as the number of frequent patterns is significantly large. The problem of extracting knowledge from infrequent patterns is far more complex as the number of infrequent patterns is even larger than that of frequent patterns, in some applications.

This chapter introduces two methods for extracting useful patterns for outlier detection. The first one is based on traditional RST, where the patterns are found based on the decision table DT. This study contributes by improving the procedure for finding useful patterns for outlier detection by introducing another promising approach, called the weighted decision table WDT. The number of patterns in both the DT and WDT are much lower than the number of frequent pattern, as fewer candidate items need to be generated. Also, DT and WDT require a single pass over the data, whereas frequent patterns require multiple passes over the data to generate the candidate items.

4.2 Pattern from Decision Table

The philosophy of RST is reliant on the assumption that with every object in the universe there is a certain level of indiscernibility affecting some information (data, knowledge). When using RST, users can describe knowledge in information tables Pawlak and Skowron [2007b]or multi-tier structures Li [2007]; Li and Zhong [2003]; Yang et al. [2008]. Additionally, users can represent associations among

Table 4.1: A relational table

Object (*Transaction*)	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_1	1	1	0	0	0	0	0
t_2	0	0	1	1	0	1	0
t_3	0	1	0	1	0	1	1
t_4	0	1	0	1	0	1	1
t_5	1	1	0	0	0	1	1
t_6	1	1	0	0	0	1	1
t_7	0	1	0	1	0	1	1
t_8	1	1	0	0	0	1	1
t_9	0	0	1	1	0	1	0
t_{10}	0	0	1	1	1	1	0
t_{11}	1	2	2	0	1	0	2
t_{12}	2	1	1	2	2	2	0
t_{13}	0	2	1	2	2	2	2

data. Generally, in a decision table, every object or record is defined according to a set of attributes, and so those with the same attributes values can be grouped together in a decision table DT which called granules or patterns. These granules have different degrees of support which can be applied to measure the frequency of the pattern in the dataset. More formal properties of the decision table DT are previously discussed in Section 2.4.3.

4.2.1 Approximation of Possible Outlier Patterns in DT

The use of a DT as discussed in Section 2.4.3 provides users with a clear insight into the density of objects in a dataset, because all the objects are grouped in granules in a decision table DT, with support sup. Based on the granules' support, a user can distinguish between those granules (patterns) with possible outlier data from those patterns that are unlikely to hold outlier values.

Table 4.1 shows thirteen records that are compressed in DT Table 4.2 and result in eight different granules g_1, g_2,g_3,g_4,g_5,g_6,g_7 and g_8 with different support

Table 4.2: A decision table

G	*Set of Attributes*							*sup*	*coverset*
	Condition					*Decision*			
	a_1	a_2	a_3	a_4	a_5	a_6	a_7		
g_1	0	0	1	1	1	1	0	1	$\{t_{10}\}$
g_2	1	1	0	0	0	0	0	1	$\{t_1\}$
g_3	0	0	1	1	0	1	0	2	$\{t_2, t_9\}$
g_4	0	1	0	1	0	1	1	3	$\{t_3, t_4, t_7\}$
g_5	1	1	0	0	0	1	1	3	$\{t_5, t_6, t_8\}$
g_6	1	2	2	0	1	0	2	1	$\{t_{11}\}$
g_7	2	1	1	2	2	2	0	1	$\{t_{12}\}$
g_8	0	2	1	2	2	2	2	1	$\{t_{13}\}$

sup: 1,1,2,3,3,1,1,1 respectively. Then, based on Hawkins definition for outlier data, we can conclude that outlier objects are not going to arise in granules with high levels of support, such as g_3, g_4, and g_5; this is because they are frequent. We utilise the weight of support to classify granules into two sets: high frequency granules and low frequency granules, based on Equation 4.1.

$$avg_{sup} = \frac{1}{|DT|} \sum_{g \in DT} \sup(g_i) \tag{4.1}$$

The average support *avg_sup* minimises the searching space, as the outlier objects are unlikely to appear among sets of high frequency granules.

Definition 4. ***(Distinguish between high frequent and low frequent granules)*** *Let DT be a decision table and G be the set of granules. The set of low frequent granules,* L_G=$\{g \in G, sup(g) < avg\_sup\}$ *and the set of high frequent granules,* H_G=$\{g \in G, sup(g) \geqslant avg\_sup\}$

We refer to g_i as a possible outlier granule or low frequent granule if its sup_g

$< avg\_sup$, the average support. Recall from the previous example, the possible granules outliers using the above formula are $g_1$and g_2 as their sup_g are less than the 2, the avg_sup.

The property of definition 4 has two advantages. The first one is that we approximate the location of possible outlier granules. This will have significant advantages when reducing the running time as the algorithm is not needed to mine the H_G granules and calculate the distance between the high frequency granules $|H_G|\times|H_G|$ because H_G granules are unlikely to hold outlier objects. Additionally, this study considers the granules to be able to detect outliers. Also, definition 4 eliminates the impact of the user's involvement in specifying parameters.

4.3 Weighted Decision Table

The use of the decision table DT, as discussed in Section 2.4.3, provides users with a clear insight into the density of objects in a dataset because all objects are grouped in granules in decision table DT with different supports (sup). Support is a useful indication to test whether a granule is frequent or infrequent. However, although after utilising the approximation of possible outlier patterns, the number of outlier granules is still large. Additionally, these possible outlier patterns have an equal degree of support and therefore, it becomes difficult to detect outlier granules. For example, if the specified threshold for a support was 1 in DT Table 4.2, then we have five possible outlier candidates with an equal chance of being outliers. To determine the deviation of the these five granules, users need to compute the distance for each one from its neighbours in order to measure the similarity or the degree of deviations to the reminder of the granules and assign

the outlier degree accordingly. This solution is expensive and becomes intractable with expansion of data size and dimensionalities.

These limitations in DT have motivated the introduction of a new useful extraction pattern, called the Weighted Decision Table WDT in this research. The WDT approach differs from that of the traditional DT. Firstly, we can disregard the assumption of finding the function between attributes in a decision table. Secondly, the WDT does not mainly rely on the find frequent item sets as described in Table 4.2. Instead, a WDT computes the weight of those item sets that share the same attributes values. This new approach provides more meaningful knowledge than the existing decision table because it considers the weight of each value in the information table, besides the covering set.

4.3.1 Pattern Extraction based on WDT

This part illustrates the properties of the WDT that make it suitable for extracting useful patterns. The following is a formal definition of the WDT.

Definition 5. ***(Weighted decision table)*** *Let $DT = (T, V^T, C, D)$ be a decision table and G be the set of granules. The WDT weighted decision table is a n-by-m matrix $WDT_{n\times m}$ where $n =\mid G \mid$ and $m =\mid V^T \mid$.*

The 4.2 computes the WDT for each item.

$$WDT_{ij} = \sum_{g\in G, g(a_j)=g_i(a_j)} sup(g) \tag{4.2}$$

for all $1 \leq i \leq n$ & $1 \leq j \leq m$

The Algorithm 1 replaces the value for the granules in DT to its corresponding weight, as found by Equation 4.2. The weighting in the W DT method is based

on computing the frequency (or the support) for the granules. After converting the DT into a WDT using Equation 4.2, we can obtain more meaningful and useful information, including degree of support *sup*, total weight TW, and actual patterns and covering sets of records.

Algorithm 1: Weighted Decision Table

Input : $DT = (T, V^T, C, D)$ - a decision table
Output: WDT A weighted decision table and TW

```
Let G = T/V^T; //G is set of granules in DT table
for g_i in G do
    Let TW_gi = 0 //Total weight
    for a_j in V^T do
        WDT_ij = Σ_{g∈G, g(a_j)=g_i(a_j)} sup(g);
        TW_gi = TW_gi + WDT_ij;
```

Table 4.3 illustrates the extraction of weighted patterns from the decision table patterns DT in Table 4.2.

The use of $WDT_{n\times m}$ matrix in a weighted decision table, WDT enables users to reduce the size of possible outlier patterns.

Table 4.3: A Weighted decision table (WDT)

G	*Set of Attributes*							*sup*	*total* weight	*coverset*
	a_1	a_2	a_3	a_4	a_5	a_6	a_7			
g_1	7	3	5	6	2	9	5	1	37	$\{t_{10}\}$
g_2	5	8	7	5	9	2	5	1	41	$\{t_1\}$
g_3	7	3	5	6	9	9	5	2	44	$\{t_2, t_9\}$
g_4	7	8	7	6	9	9	6	3	52	$\{t_3, t_4, t_7\}$
g_5	5	8	7	5	9	9	6	3	49	$\{t_5, t_6, t_8\}$
g_6	5	2	1	5	2	2	2	1	19	$\{t_{11}\}$
g_7	2	8	5	2	2	2	5	1	26	$\{t_{12}\}$
g_8	7	2	5	2	2	2	2	1	22	$\{t_{13}\}$

4.3.2 Approximation of Possible Outlier Pattern in WDT

Once we have computed the TW in the WDT, users can utilise the TW values to approximate possible outlier granules. Therefore, Equation 4.3 separates frequent patterns from infrequent ones.

$$avg\_weight = \frac{1}{|G|} \sum_{g \in WDT} TW(g) \tag{4.3}$$

where the total weight is calculated from Equation 4.4.

$$total\ weight(g_i) = \sum_{a_j \in V^T} WDT_{ij} \tag{4.4}$$

Definition 6. ***(WDT for possible outlier patterns)*** *Let WDT be a Weighted decision table and G be the set of granules. The possible outlier patterns $g_i \in L_G$ in $WDT_{n \times m}$ matrix must satisfy the following:*

$$total\ weight(g_i) < avg\_weight$$

The patterns that have a TW lowest than the avg_weight are possible candidates to hold outlier values and groups in the set of L_G granules. The patterns with $TW \geqslant avg_weight$are unlikely to include outlier data which are grouped in high granule set H_G.

Algorithm 2 describes the procedure for approximating possible outlier patterns.

Based on the algorithm 2, the number of possible candidate outlier patterns L_G found in Table 4.3 are three. These patterns are g_6, g_7, g_8 which reflect to records t_{11}, t_{12} and t_{13} respectively because their total weight is less than the 36.25

Algorithm 2: Approximation of Possible Outlier Pattern

Input : WDT, TW - a weighted decision table and total weight
Output: L_G - Possible outlier granules

Let $G = T/V^T$; //G is set of granules in WDT table
Let $avg\_weight = \frac{1}{|G|} \sum_{g \in WDT} TW(g_i)$
$L_{G[i]} = \emptyset$;
for g_i *in* G **do**
    if $TW(g_i) < avg\_weight$ **then return**
    $L_{G[j]} = L_{G[j]} \cup \{g_i\}$;
    ;
Return All candidate outlier granules in L_G

which is average total weight. By comparing it with the approximation of possible outlier patterns found by DT, the WDT has a lower number of possible outlier patterns than DT. For example, according to Table 4.2, there are 5 candidate patterns found by DT, but according to new WDT in Table 4.3, there are only 3 outlier patterns.

4.4 Chapter Summary

This chapter has explained how to extract a useful pattern based on RST. It introduces two approaches for mining patterns, the decision table, based on traditional RST, and the WDT, which computes the weight of the items in the DT. The items in the DT are grouped based on granularity (or similarity) generating number of granules. The support degree sup in the DT determines the frequency of the pattern. This study utilises the degree of support to distinguish between normal patterns HG and possible outlier patterns LG. WDT brings significant advantages, particularly in an outlier study. However, both DT and WDT make fundamental contributions to both outlier detection and quality assessment as

will be discussed in the following chapters.

Chapter 5

Algorithms for Outlier Detection

5.1 Introduction

The method of extracting possible candidate patterns described in Chapter 4 represents a significant breakthrough in mining and finding approximate possible outlier patterns for two reasons. The first is that mining patterns by DT and WDT as discussed in Chapter 4 is much faster than frequent pattern mining (Apriori Algorithm). Secondly, fewer extracted patterns are found by DT and WDT than by frequent pattern mining. The approximation algorithms for both DT and WDT further reduce the mining space for outlier data. These advantages of DT and WDT enable the design of three different effective outlier detection algorithms.

This chapter introduces three interesting outlier detection algorithms. The first algorithm is called Granule-Based Outlier Detection (GBOD). This algorithm mines outlier patterns from the approximate patterns found by DT as described in Chapter 4. The second outlier algorithm is the Ranking Weighted

Decision Table (RWDT) algorithm. The last outlier algorithm is the Centroid Granule (CG) algorithm. Both the RWDT and CG algorithms mine the patterns found by WDT so as to detect outlier data.

5.2 Granules Based Outlier Detection

This section provides a detailed analysis of granule based outlier detection (GBOD). The proposed GBOD algorithm captures and measures the degree of outlier objects.

The use of the decision table discussed in Chapter 4 provides users with a clear insight into the density of objects in a dataset, because all objects are grouped in granules in decision table DT with different degrees of support sup. Based on the approximation of finding candidate outlier patterns, the user can differentiate between normal patterns which have high frequency HG and the low frequency patterns LG, which are the possible outlier candidates. The average support in Equation 4.1 classifies granules into two sets: high frequency HG and low frequency LG. The average support avg_sup minimises the search space as the outlier objects are unlikely to appear amongst the set of high frequency granules.

We can recall from Chapter 4, Table 4.2, which shows that there are 8 granules in DT: g_1, g_2,g_3,g_4,g_5,g_6,g_7 and g_8 with different support sup: 1,1,2,3,3,1,1,1 respectively. The candidate patterns L_G in Table 4.2 are g_1, g_2,g_6,g_7 and g_8.

The GBOD algorithm considers that not all LG granules are outliers. Hence, this algorithm introduces the weighted discernibility matrix approach to measuring outlier degrees for each LG granule and ranks them in descending order.

5.2.1 Discernibility Matrix Approach

The discernibility matrix is commonly used in many rough set applications. It is a symmetric matrix that measures the difference between objects in the information system $|U| \times |U|$, where (U is a set of objects) or in the decision table $|G| \times |G|$, where G is a set of granules (see Rauszer and Skowron [1992]; Skowron and Synak [2004]).

Definition 7. ***(Discernibility Matrix)*** *Given a decision table DT= (G,A), where G is set of granules and A= { a_1, a_2, ..., a_m } set of attributes. Let g_i and $g_j \in G$ be two granules, the discernibility matrix is defined as follow:*

$$G_{ij} = |\{a \in A : a(g_i) \neq a(g_j)\}| \;\; i, j = 1, ..., |G|\,,$$

where G_{ij} be the cell in ith row and jth column.

From the above definition, it is clear that the discernibility matrix counts the number of attributes in A for which granules g_i and g_j have different attribute values. For example, if the objects g_i and g_j have different values for a correspondent attribute, then the difference between g_i and g_j is 1. Otherwise, the difference between g_i and g_j is 0.

$$total - difference(g_i) = \sum_{g=1}^{|G|} G_{ij} \tag{5.1}$$

Table 5.1 illustrates the discernibility matrix for computing the distance between granules found in Table 4.2 in Section 4.2. For example, the total different

Table 5.1: Discernibility Matrix (DM)

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
g_1	0	6	1	4	6	6	5	5
g_2	6	0	5	4	2	4	5	7
g_3	1	5	0	3	5	7	5	5
g_4	4	4	3	0	2	7	6	6
g_5	6	2	5	2	0	5	6	6
g_6	6	4	7	7	5	0	7	5
g_7	5	5	5	6	6	7	0	3
g_8	5	7	5	6	6	5	3	0

Table 5.2: Top 5 Outlier Result Based on Discernibility Matrix

Outlier Granule	*Outlier Record*	$\sum$*Distance*
g_6	t_{11}	41
g_7	t_{12}	37
g_8	t_{13}	37
g_1	t_{10}	33
g_2	t_1	33

between g_1 and g_2 in Table 5.1 is 6. This means that there are six attributes having different values in g_1 and g_2 and only one attribute that has the same value in both g_1 and g_2. Table 5.1 shows all the different between granules. The user can sum these figures using Equation 5.1 to determine the granule that is furthest away from the other granules. Table 5.2 shows the Top 5 granules with highest total deference and their associated records. It is clear from Table 5.2 that g_6 is the strongest outlier granule found by DM approach because its total deference from other granules is the highest with degree 41.

The discernibility matrix is undoubtedly useful in calculating the distance between objects or granules if the pair have equal attribute weight. However, in most information systems, weights for attributes are different. For example,

Table 5.2 shows that g_1 and g_2 have equal difference. This might produce inaccurate detection and ranking for outlier data. Hence, it would be more accurate and meaningful if the user could use the weight and the discernibility matrix to measure the difference between a pair of granules. The next section introduces a new Weighted Discernibility Matrix Approach for outlier detection. The proposed GBOD approach computes the similarities and deviations between H_G and L_G granules using a new weighted discernibility matrix (WDM).

5.2.2 Weighted Discernibility Matrix Approach

The GBOD algorithm modifies the original definition described above for the DM matrix and introduces a new matrix called the Weighted Discernibility Matrix (WDM). In the WDM, the weight for each attribute is considered when we calculate the differences between pairs of granules. WDM $(|G| \times |L_G|)$ calculates the difference between $|L_G| \times |G|$ where G is a set of all granules in DT and L_G is a set of possible candidate outliers in DT.

Definition 8. ***(Weighted Discernibility Matrix, (WDM))*** *Given a decision table $DT = (G, A)$, where G is a set of granules and $A = \{a_1, a_2, ..., a_m\}$, a set of attributes. Let G be all granules in DT and $L_G \subseteq G$, with low support (e.g., $\forall\ g_i \in L_G$, $sup(g_i) <$ average support avg_{sup}. The weighted discernibility matrix between pair granules g_i and $g_j \in DT$ is calculated as follows:*

$$WDM_{ij} = G_{ij} \times \frac{sup(g_j)}{\sum\limits_{g_j \in G} \sup(g)} \tag{5.2}$$

Table 5.3: Weighted Discernibility Matrix (WDM)

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
g_1	0	$6 \times \frac{1}{13}$	$1 \times \frac{2}{13}$	$4 \times \frac{3}{13}$	$6 \times \frac{3}{13}$	$6 \times \frac{1}{13}$	$5 \times \frac{1}{13}$	$5 \times \frac{1}{13}$
g_2	$6 \times \frac{1}{13}$	0	$5 \times \frac{2}{13}$	$4 \times \frac{3}{13}$	$2 \times \frac{3}{13}$	$4 \times \frac{1}{13}$	$5 \times \frac{1}{13}$	$7 \times \frac{1}{13}$
g_6	$6 \times \frac{1}{13}$	$4 \times \frac{1}{13}$	$7 \times \frac{2}{13}$	$7 \times \frac{3}{13}$	$5 \times \frac{3}{13}$	0	$7 \times \frac{1}{13}$	$5 \times \frac{1}{13}$
g_7	$5 \times \frac{1}{13}$	$5 \times \frac{1}{13}$	$5 \times \frac{2}{13}$	$6 \times \frac{3}{13}$	$6 \times \frac{3}{13}$	$7 \times \frac{1}{13}$	0	$3 \times \frac{1}{13}$
g_8	$5 \times \frac{1}{13}$	$7 \times \frac{1}{13}$	$5 \times \frac{2}{13}$	$6 \times \frac{3}{13}$	$6 \times \frac{3}{13}$	$5 \times \frac{1}{13}$	$3 \times \frac{1}{13}$	0

Since GBOD relies on DT for finding patterns and approximating possible outlier patterns, the WDM will utilise the DT patterns found in Table 4.2 in Section 4.2 and the approximation in Section 4.2.1. In these sections there are five possible outlier granules there are five possible outlier granules L_G (g_1, g_2,g_6,g_7,g_8) and three normal granules H_G (g_3, g_4,g_5). The GBOD algorithm measures the distance between all G, which includes H_G and L_G and L_G, as illustrated in Definition 8.

Table 5.3 provides an example of the new WDM matrix. Here, the value of $WDM_{1,4}$ calculated using the WDM matrix is $(0+1+1+0+1+0+1) \times \frac{3}{13}$ where 3 is the support of g_4 and 13 is the total support for all G granules. The strongest outlier granule in WDM matrix is the granule that has the farthest distance from the other granules.

After calculating the distance between L_G and G granules, the user can easily calculate the $GBOD$ outlier degree for entire L_G set by using Equation 5.3.

$$disc(gi) = \sum_{g_j \in G} WDM_{ij} \tag{5.3}$$

The user can retrieve the Top K outlier granules in descending order from the

Table 5.4: Top 5 Outlier Result Based on GBOD algorithm

Outlier Granule	*Outlier Record*	$\sum$*Distance*
g_6	t_{11}	5.54
g_7	t_{12}	5.07
g_8	t_{13}	5.07
g_1	t_{10}	4.15
g_2	t_1	3.84

strongest to the weakest. Table 5.4 shows the Top 5 outlier granules and their associated records, found by the GBOD algorithm using the WDM. In Table 5.4, the GBOD algorithm flags g_6 as the strongest outlier granule with a distance of a degree 5.54

The results of outlier detection using GBOD compared with using the traditional DM prove the effectiveness of GBOD for outlier detection. However, there are other advantages to using the WDM with the GBOD algorithm rather than the traditional DM. The first advantage is that the number of granules in the WDM is smaller than in the DM as the WDM computes the distance between LG and G whereas DM computes the distance between G and G. The second critical advantage of using the WDM is that the ranking for outlier data in the WDM is more accurate than in the traditional DM. Tables 5.2 and 5.4 show the Top 5 outlier patterns. However, it is difficult to correctly rank the outlier granules in the DM. For example, the granules g_1 or g_2 have the same outlier score with degree 33. Unlike the traditional DM, the use of the GBOD algorithm with the new WDM has provided more accurate results with regard to ranking. The GBOD algorithm distinguishes between g_1 and g_2 the distances of these granules being 4.15 and 3.84 respectively (see Table 5.4). This makes the ranking of outlier data more accurate. Algorithm 3 describes the procedure of the GBOD

algorithm.

Algorithm 3: GBOD Algorithm Based on WDM Matrix

Input : L_G, G - a set of possible outlier granules and granules in DT
Output: $Top-K$ - Top-K outlier granules

Calculate the DM matrix $G_{n\times m}$ using Definition 7
for $G_{ij} \in G_{n\times m}$ **do**
    $WDM_{ij} = G_{ij} \times \frac{sup(g_j)}{\sum_{g_j \in G} \text{sup}(g)}$;
for $g_i \in L_G$ **do**
    $disc(gi) = \sum_{g_j \in G} WDM_{ij}$;
Sort L_G in descending order based on $disc(gi)$
Return Top-K granule outliers

5.3 Ranking Weighted Decision Table for Outlier Detection

This chapter presents a promising solution for outlier detection. The GBOD algorithm described above provides an effective solution for the detection of outlier data. However, the main limitation of the GBOD algorithm is that the number of possible outlier patterns in L_G is still large. The reason for this is that patterns found by GBOD are based on the frequency of granules in DT where the support degree is the basis for classification of patterns into H_G and L_G.

For example, Figure 5.1 shows the patterns distribution found by the DT. The example in Figure 5.1 illustrates the three common patterns in most KDD datasets: Frequent Patterns, Uncertain patterns, and Outlier patterns. If the specified min-sup was 3, then the first 20 patterns are considered frequent patterns H_G because their support ≥ 3. This reduces the mining space for outlier points

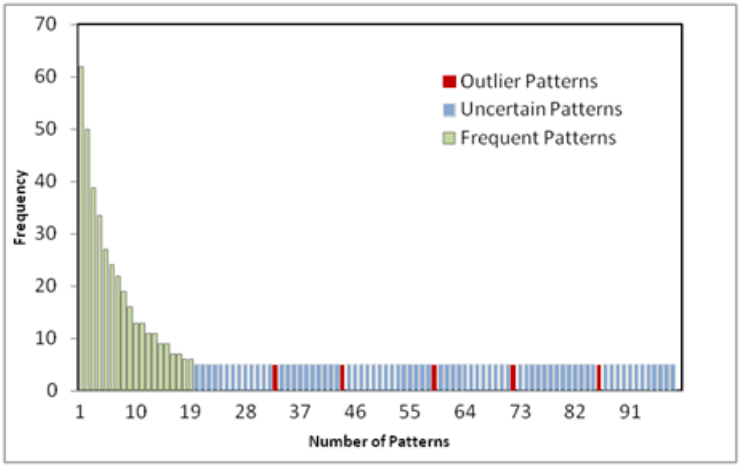

Figure 5.1: Pattern Distribution

as outlier patterns do not appeared amongst frequent patterns. However, the number of infrequent patterns L_G which includes outlier patterns and uncertain patterns is still significantly large. The problem in finding frequent items based on DT or any other frequent items FIs algorithms is that users are unable to examine whether or not the items inside patterns are frequent or infrequent. In an uncertain pattern, most of its items are frequent except a few items. Whereas, most items in an outlier pattern are infrequent.

Based on this observation, the goal of RWDT is to address the problem of how the user can effectively expose outlier patterns and re-shuffle the patterns in an ascendant curve to clearly classify patterns into three groups: outlier patterns, uncertain patterns and frequent patterns. The RWDT for mining outlier data relies on the WDT technique, discussed in Chapter 4 for finding useful outlier patterns.

This section illustrates the capabilities of the RWDT algorithm in detecting outlier data without computing the distance between patterns to determine the

Table 5.5: RWDT for object outlier detection

G	*Set of Attributes*							*sup*	*total* weight	*coverset*
	a_1	a_2	a_3	a_4	a_5	a_6	a_7			
g_6	5	2	1	5	2	2	2	1	19	$\{t_{11}\}$
g_8	7	2	5	2	2	2	2	1	22	$\{t_{13}\}$
g_7	2	8	5	2	2	2	5	1	26	$\{t_{12}\}$

degree of deviations or similarities, and in exposing different types of outlier, including outlier patterns, objects and items.

5.3.1 RWDT for Object Outlier Detection

The proposed WDT has great advantages over DT for finding useful possible outlier candidates which were discussed in detail in Section 4.3. The use of the $WDT_{n\times m}$ matrix and the total weight TW in Section 4.3 enables users to easily apply the RWDT algorithm. The RWDT algorithm uses TW to rank the patterns. Based on outlier detection, which defines outlier data as infrequent points that significantly differ from the remainder data, the strongest outlier pattern is the pattern with the lowest TW ranked by the RWDT algorithm.

To enhance the RWDT algorithm and make the process of sorting patterns more efficient , RWDT deploys Algorithm 2 described in Section 4.3.2. The number of patterns in L_G set found by Algorithm 2 is small. Therefore, the ranking procedure in the RWDT algorithm 4 is more efficient compared with the GBOD algorithm described in Section 5.2 and is comparatively effective for flagging outlier data.

Table 5.5 shows the ranked outlier patterns with their correspondent objects from WDT, taken from Table 4.3 in Section 4.3.1. From the Table 5.5, the possible outlier granules are g_6, g_7, g_8 which reflect to objects t_{11}, t_{12} and t_{13}

respectively because their total weight are less than average total weight of 36.25. The strongest outlier, based on RWDT, is record t_{11} the corresponding granule g6 has a total weight TW of 19 which is the lowest total weight. Algorithm 4describes the details of the RWDT outlier detection algorithm.

Algorithm 4: RWDT Algorithm for Object Outlier Detection

Input : G,L_G, TW - a set granules and of possible outlier granules and total weight

Output: OG - Ranking outlier granules

1 Let $avg\_weight = \frac{1}{|G|} \sum_{g \in WDT} TW(g)$

2 $OG = \emptyset$;

3 **for** $g_i \in L_G$ **do**

4 **if** $TW(g_i) < avg\_weight$ **then return**
$OG = TW(g_i) \cup \{g_i\}$;
;

5 Sort OG in descending order based on TW;

The properties presented by RWDT show the capabilities of the proposed algorithm in detecting both outlier patterns and the covering set which indicates the location of the objects (or records). The RWDT method takes into consideration that there are some attribute values that make g_i behave as an outlier. Hence, in the following section we extend RWDT to introduce RWDT for attribute outlier detection.

5.3.2 RWDT for Attribute Outlier Detection

In this section, we show how RWDT can detect a set of attributes that cause a granule to be an outlier. Because RWDT only uses attribute weights, we can easily calculate the average attribute weight for each granule using Equation 5.4.

Table 5.6: A Set of outlier attributes in RWDT

G	*Set of Attributes*							*coverset*	total weight	average attribute weight	Outlier Attribute
	a_1	a_2	a_3	a_4	a_5	a_6	a_7				
g_6	5	2	1	5	2	2	2	$\{t_{11}\}$	19	2.7	a_2,a_3,a_5,a_6,a_7
g_7	2	8	5	2	2	2	5	$\{t_{12}\}$	26	3.7	a_1,a_4,a_5,a_6
g_8	7	2	5	2	2	2	2	$\{t_{13}\}$	22	3.1	a_2,a_4,a_5,a_6,a_7

$$avg\_attr\_weight(g_i) = \frac{1}{|V^T|} \sum_{a_j \in g_i} WDT_{ij} \tag{5.4}$$

Equation 5.4 can clearly expose attribute outliers since they tend to be further away from the average attribute weight. Hence, the stronger an outlier granule g_i, the more likely it is to have many outlier attributes a_j that have a weight lower than the average attribute weight.

Definition 9. ***(WDT for attribute outlier detection)*** *Formally, let OG be all outlier objects and $g_i \in OG$. Attribute a_j is an outlier attribute for g_i if*

$$WDT_{ij} < avg\_attr\_weight(g_i)$$

For example, Table 5.6 shows the outlier granules that were found in Section 5.3.1 along with the set of outlier attributes found by Algorithm 5. Based on total weight, granule g_6 is the strongest outlier. When we investigate the reason for this, we find that there are five attributes: a_2,a_3,a_5,a_6 and a_7 whose weight occur infrequently and are far away from the average attribute weight. The covering set and set of attributes columns in Table 5.6 can be efficiently used to reveal the identity of the outlier record and the actual attribute values.

It is clear from the above discussion that the properties of RWDT, both for RWDT for object outlier detection and RWDT for attribute outlier detection, are

Algorithm 5: RWDT for Attribute Outlier Detection

Input : WDT, OG, V^T - WDT matrix, Outlier granules, Attributes
Output: $OA[1, ..., |OG|]$, each $OA[i]$ is a set of outlier attributes for granule g_i

```
for g_i ∈ OG do
    avg_attr_weight(g_i) = 1/|V^T| Σ_{a_j ∈ g_i} WDT_ij;
    OA[i]= ∅;
    for a_j ∈ V^T do
        if WDT_ij < avg_attr_weight(g_i) then return
        OA[i] = OA[i] ∪ {a_j};;
Return OA;
```

interesting and provide a promising solution for outlier detection. Hence, in this research we will extensively validate the RWDT method by using many datasets and comparing its effectiveness and efficiency with different state-of-art methods

5.4 Centroid Granule for Outlier Detection

Most contributions in data quality research involve non-parametric methods. These include distance-based methods , density-based methods and cluster-based methods. Most of these methods compute the distance between a point and its nearest neighbour points. Figure 5.2 illustrates a 2-dimensional dataset (X,Y) with 502 objects, where the Top 10 outlier objects are highlighted. The distance of each of these 10 points to the correspondent nearest neighbour points is higher than the specified minimum distance, and therefore these 10 points are labelled as outlier points.

However, these methods encounter serious limitations for mining outliers in a large dimensional dataset. The first limitation is that these methods rely on

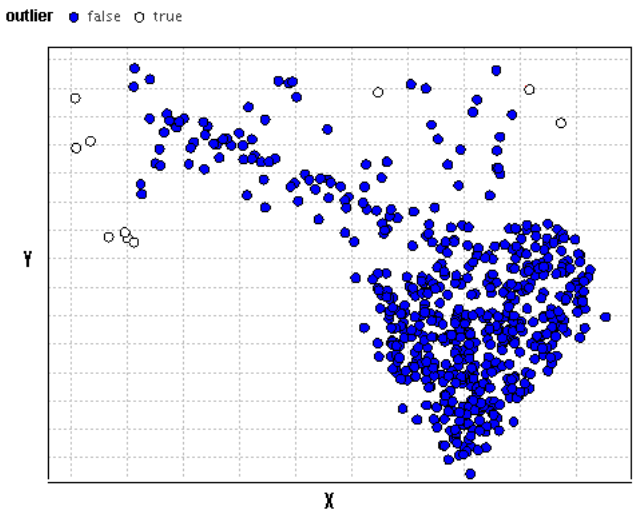

Figure 5.2: Top 10 Outlier Objects

a distance based approach with a quadratic time complexity, since users need to calculate distances between many pairs of objects. The time complexity of these methods will significantly increase with increasing data size and dimensions (attributes), leading to solutions that can be expensive and intractable.

The second serious problem of existing outlier methods based on distance is the sensitivity to the specified parameters. It is hard for the user to specify an adequate minimum distance (or threshold). If the specified threshold is too large, users may miss some true outlier objects. Conversely, if the specified threshold is too small, users may get a lot of objects falsely labelled as outliers. Similar problems occur when users specify the number of nearest neighbour points. If the specified number of nearest points is too small, the algorithm might miss true outlier points or incorrectly flag outlier points. Conversely, if the specified number of nearest points is too large, then the algorithm has serious running time problems reducing its efficiency for mining outliers from a large dimensional dataset.

The proposed Centroid Granule for Outlier Detection (CG) algorithm has addressed these limitations discussed above in a very efficient and effective manner. The properties of the proposed CG algorithm, discussed in next section, prove its capabilities in mining outlier data in a large dimensional dataset.

5.4.1 Finding the Centroid and Outlier data

The properties of WDT and the approximation of possible outlier patterns, described in Sections 4.3 and 4.3.2 respectively, play an essential roly in the CG algorithm. Algorithm 2 in Section 4.3.2 distinguishes between frequent H_G and infrequent L_G patterns. The problem is how a user can efficiently measure the distance between L_G patterns and the remainder data so that the user can return the strongest outlier data from the L_G in ranked order.

The proposed CG algorithm efficiently and effectively solves the problem associated with the distance based approach by computing the centroid weighted granule. Since the patterns in WDT are based on computing the weight of items, the user can easily compute the CG point. This has a great advantage with regards to running time as the algorithm will only compute the distance of L_G patterns to the centroid granule rather than measuring the distance to many objects.

Definition 10. ***(Centroid Granule CG)*** *Let* $WDT = (T, V^T)$ *be a matrix n-by-m* $WDT_{n\times m}$ *where* $n =\mid G \mid$ *and* $m =\mid V^T \mid$ *in weighted decision table and* G *be the set of granules. A centroid granule* CG *is a vector of* $\langle CG_1, ..., CG_m\rangle$ *where the centroid weighted* CG_j *for attribute* $a_j \in V^T$ *for All* G *is calculated as follows:*

$$CG_j = \frac{1}{|G|} \sum_{g_i \in G} WDT_{ij} \tag{5.5}$$

The Algorithm 6 describes the process of outlier detection using the centroid granule. Firstly, the Algorithm 6 finds the centroid granule for all granule in WDT. Then, it utilises the Euclidean distance to compute the distance between the set of possible outliers OG and the centroid granule CG. Finally, it returns the top outlier granules from OG sorted in descending order based on distance to the centroid granule CG.

Algorithm 6: Centroid Granule Algorithm

Input : WDT - a weighted decision table
Output: OG - Top outlier granules

Let L_G; //A set of possible outlier granules
for $a_j \in V^T$ **do**
    $CG_j = \frac{1}{|G|} \sum_{g_i \in G} WDT_{ij}$;
for $g_i \in L_G$ **do**
    $dis(g_i, CG) = \sqrt{\sum_{a_j \in V^T} [a_j(gi) - CG_j]^2}$;
Sort OG in descending order based on $dis(g_i, CG)$
Return top outlier granules in OG

The properties of the proposed centroid granule CG have several advantages. In particular, using the approximation of possible outlier granules and the centroid granule, presented in Definitions 6 (see Section 4.3.2) and 10 respectively, can significantly reduce running time.

5.5 Chapter Summery

Outlier detection is a critical component of many applications including data mining, credit card fraud detection and other detection. Detecting abnormal behaviour can have not only financial benefits associated with fraud detection or improving the accuracy of mining model, but also can be more important in saving peoples lives by exposing terrorist activities. This chapter presents three different outlier algorithms: GBOD, RWDT and CG. Each one of these algorithms solves a critical problem in mining outlier data. GBOD introduces a new approach based on the weighted discernibility matrix. The proposed GBOD algorithm proves that the use of the weighted discernibility matrix can be as accurate as the use of Euclidean distance with regard to detecting outlier data. The second and the third proposed outlier algorithms, RWDT and CG respectively, provide efficient solutions for detecting outlier data. The RWDT algorithm reorders the patterns in a way that ranks the strongest outlier patterns first without computing the distance between patterns. The CG algorithm determines the centroid pattern and measures outlier degree by calculating the distance from each point to the centroid granule rather than the distance to a set of nearest neighbour patterns or points. The contributions of these proposed outlier algorithms are not only useful for providing effective outlier detection methods but also providing efficient solutions in mining outlier data from large dimensional databases.

Chapter 6

Quality Assessment

6.1 Introduction

One big challenge in data quality research is how best to assess the quality of data in a database or data warehouse. The preliminarily focus of data quality literature is on detecting and correcting poor data, such as outliers and incomplete and inaccurate data. This narrow view of data quality problems abstracts the move towards a complete automated data quality solution. Data errors are continuously occurring and being stored again and again in databases and data warehouses.

Quality assessment is a critical step for providing a complete data quality solution. "Nothing is more likely to undermine the performance and business value of a data warehouse than inappropriate, misunderstood, or ignored data quality " Ballou and Tayi [1999]. A well known study estimates that the immediate cost stemming from a 1-5 percent error rate is approximately 10% of revenue Redman [1998]. In the U.S., for instance, data quality problems cost U.S. businesses more than 600 billion dollars per year Batini and Scannapieco [2006]. Several studies,

as discussed in the literature review chapter of this study, draw attention to data quality and investigate the impact of poor data on customer satisfaction, decision making, operational costs, and executing strategy Ballou and Pazer [1985]; Lee et al. [2002]; Redman [1996]; Strong et al. [1997]; Wand and Wang [1996]; Wang and Strong [1996]. However, the existing contributions of quality assessment are inadequate for providing a complete quality assessment solution.

To end this, researchers and practitioners need to take a broader view to include other essential aspects of data quality assessment. In particular, there are two aspects that must be addressed in quality assessment. The first one is how to assess quality changes in the data. The second aspect is how to identify the location of the most severe data errors. Having a proper quality assessment method that efficiently and effectively addresses these aspects will facilitate a significant breakthrough in automated data quality assessment. Users will be able to capture quality change in the information systems, maintaining high data quality, redesigning organisational processes in the way that error data are captured prior to accessing the information systems and specifying and allocating the right resources for conducting quality improvement tasks.

6.2 Motivation Example

Existing quality assessment fails to provide a reliable solution that assesses quality change and identifies the location of the most severe data errors. For assessing quality change, the most common quality assessment measurement relies on error rate or accuracy rate. Before applying error rate or accuracy rate, all data in the relation matrix must be converted to a binary matrix where 0 represents normal

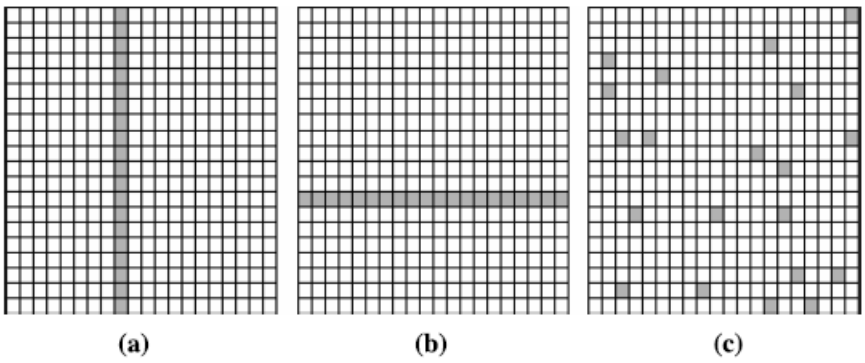

Figure 6.1: Error Distributions Fisher et al. [2009]

data and 1 represents error data. Error rate can then be applied to the binary matrix by dividing the number of defective data fields by the total number of fields. Alternatively, the user can compute the accuracy rate by subtracting the error rate from 1 (1-error rate). Both error rate and accuracy rate are useful to show the rate of defective or accurate data in a database. This often causes interpretation problems. Error rate or accuracy rate is likely to provide users with the same results particularly when users want to assess quality status (improvement or degradation) in a database across various times, which could lead to an inaccurate conclusion. For example, Figure 6.1 shows three databases (a), (b) and (c) each with 5% of error rate for each one. Although, the error rate in databases (a), (b) and (c) are the same with 5%, the location of these errors are differs considerably between the three databases.

The second drawback of existing quality assessment methods is that users are unable to identify the location of the most severe data errors. Figure 6.1, , illustrates different error distributions for databases (a), (b) and (c). Errors in databases (a) and (b) are systematically distributed in a column and row respectively, but in (c) are randomly distributed across columns and rows. Undoubtedly,

dealing with the errors in databases (a) and (b) is going to be completely different to (c) in terms of time, cost, effort, complexity, and solutions. Measuring the degree to which error distribution is systematic or random is an essential step for quality assessment. It enables decision makers to have more insight into quality problems in the database and provides more accurate estimation of the complexity of the cleaning process.

6.3 Preliminaries

The quality assessment proposed in this section follows the same procedure as most data quality assessment methods by transforming the actual data values to a binary matrix (0 representing a normal value and 1 an outlier value). The quality assessment method is based on Rough Set Theory (RST), and in particular on the decision table DT discussed in Section 2.4.3.

This study assumes the existence of a database T. Formally, T can be described as a decision table DT_A as shown in Table 6.2 (G_i,A_i), where G_i is a set of granules about attributes A_i, and a granule is a group of objects (rows) which have the same attribute values Pawlak [1991]. Table 6.1 shows an example of a binary matrix. The values 0 refer to the normal data and the values 1 represent outlier data.

We usually assume that there is a function for every attribute $a \in A$ such that $a : T \rightarrow V_a$, where V_a is the set of all values of a. We call V_a the domain of a, for example, $V_a = \{1, 0\}$ in binary matrix. User can extract the patterns based on the DT technique discussed in Section 2.4.3. Table 6.2 shows four granules extracted from the binary matrix shown in Table 6.1, where the granule support,

Table 6.1: A relational table

Object (*Transaction*)	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_1	1	1	0	0	0	0	0
t_2	0	0	1	1	0	1	0
t_3	0	1	0	1	0	1	1
t_4	0	1	0	1	0	1	1
t_5	1	1	0	0	0	1	1
t_6	1	1	0	0	0	1	1
t_7	0	1	0	1	0	1	1
t_8	1	1	0	0	0	1	1
t_9	0	0	1	1	0	1	0
t_{10}	0	0	1	1	1	1	0

Table 6.2: A decision table

G	*Set of Attributes*							*sup*	*coverset*
	Condition					*Decision*			
	a_1	a_2	a_3	a_4	a_5	a_6	a_7		
g_1	0	0	1	1	1	1	0	1	$\{t_{10}\}$
g_2	1	1	0	0	0	0	0	1	$\{t_1\}$
g_3	0	0	1	1	0	1	0	2	$\{t_2, t_9\}$
g_4	0	1	0	1	0	1	1	3	$\{t_3, t_4, t_7\}$
g_5	1	1	0	0	0	1	1	3	$\{t_5, t_6, t_8\}$

$sup(gi)$, is the number of rows with the same values for the 7 attributes, also called the size of the covering set of the corresponding granule.

6.4 Decision Rule Method for Data Quality Assessment

The proposed decision rule method provides management with the information it needs for data quality assessment. Management will be able to determine any change in quality across different databases or data at different time periods.

This study assumes that there are two databases D1 and D2 with the same data structure, where D1 is a history database or training set; and D2 is a newly generated database or testing set. Formally, D1 (or D2) can be described as multiple decision tables DT (Gi, Ai), where Gi is a set of granules with attributes Ai, and a granule is a group of objects (rows) which have the same attribute values, as discussed in Section 2.4.3.

Table 6.1 illustrates an example of the binary matrix (0,1) where 0 represents a normal value in the original database and 1 indicates an outlier value. Using the outlier algorithms presented in Chapter 5 and in particular the RWDT algorithm, the user can build a binary matrix to assess the outlier data. The binary matrix for normal and outlier data in Table 6.1 can be compressed into granules with different support degrees, as shown in Table 6.2. Details of the method for constructing DT are given in Section 2.4.3.

Attributes A_i can be divided into two groups: condition attributes (C_i) and decision attributes (D_i), such that every granule in the decision table can be mapped into an *association rule* (or called *decision rule*), for example, the second granule, g_3, can be read as the following decision rule:

$$(a_1 = 0 \wedge a_2 = 0 \wedge a_3 = 1 \wedge a_4 = 1 \wedge a_5 = 0) \rightarrow (a_6 = 1 \wedge a_7 = 0)$$

where the antecedent and consequent are described as Boolean expressions. Referring back to our example in Table 6.2, there are five decision rules with different levels of support $sup(gi)$. Users can assign condition attributes and decision attributes according to the requirements of the user and the problem.

The decision rule method can be utilised to evaluate the quality of data from

multiple databases. Users can assign condition attributes to D1, as a history database or training set, and assign decision attributes to D2, as a newly generated database or testing set. Decision rules can be discovered from D1 and made matching in D2. The result will determine if there is a quality change in the new database. The use of the support degree in a decision rule method is useful to show the severity of poor data in a rule. If there is a quality degradation in the new dataset D2, the number of rules in D1 will not match the ones in D2, or the support for the matched rules in D2 will be significantly higher than the correspondent rules in D1.

Also, the decision rule method is useful to determine if there is a quality improvement in the newly generated data. For instance, if the number of rules in D2 is smaller than D1 and these rules match with D1, the quality of the data in D2 is improving. Another indication of quality improvement is that the ratios of support in defective rules in D2 are smaller than the corresponding ones in D1.

The following points describe the process for assessing‘ quality change using the proposed decision rule method:

- Training data (historical data)

    1. Scan the database, D1, to define the data values as normal or abnormal (defective).
    2. Transform normal data values in D1 to 0 and abnormal data values to 1
    3. Generate the corresponding decision table (G1, A) from D1 by grouping rows with the same attribute values, where A is the set of selected attributes in D1.

- Testing data (newly generated data)

    1. Process D2 in the same way as D1, and to obtain a decision table (G2, A).
    2. Compare the defective rules in decision table (G1, A) with those in decision table (G2, A); and calculate the numbers of matched and unmatched rules.
    3. For the matched rules, determine the severity of quality problem in D2 by measuring the difference in the support degree.

The above steps of the decision rule method will enable users to capture any quality change with regard to data improvement or degradation in the information system. This has a great advantage over the existing error rate or accuracy rate and p-value as the decision rule method can accurately show errant patterns as well as assessing the quality change in these patterns over time.

6.5 Randomness Degree

6.5.1 Definition for Randomness Degree

In data quality research, errors in databases are either systematically or randomly distributed across rows and columns. A systematic error presents a clear pattern of defective data. For example, errors might frequently occur in specific columns like address and zip code. On the other hand, random distribution of errors shows a lack of regularity of defective patterns. Intuitively, detecting and handling systematic errors is less complicated than detecting and handling random errors

Table 6.3: Cover All Patterns

Granule	a_1	a_2	a_3	$Sup(g_i)$
g_1	0	0	0	50
g_2	0	0	1	25
g_3	0	1	1	2
g_4	0	1	0	2
g_5	1	1	1	2
g_6	1	0	0	2
g_7	1	1	0	15
g_8	1	0	1	2

Fisher et al. [2009]. Unlike systematic errors, random errors are difficult to deal with and consume massive amounts of time and cost. The first reason is that finding a value (or values) that produces random patterns is difficult and time consuming. Secondly, the number of errant random patterns is usually large with low frequency. This makes finding potential errant patterns for solving large quality problems very difficult for decision makers to achieve.

This section introduces the randomness measure, based on patterns (or granules). Then we further enhance the proposed solution by measuring the distribution of granules by two vectors: numbers of patterns (or granules) and the pattern distribution. This enables users to expose and assign systematic and random patterns to different categories.

This section measures the randomness degree of errors, based on the numbers of errant patterns or errant granules. For example, Table 6.3 and Table 6.4 are two decision tables DT_{A1} and DT_{A2} respectively, generated from two different tables that have the same attributes "a_1,a_2,a_3" and the same size (100 rows), but with different error distributions.

As can be seen from Table 6.3 and Table 6.4, DT_{A1} has more errant granules

Table 6.4: Not Cover All Patterns

Granule	a_1	a_2	a_3	$Sup(g_i)$
g_1	0	0	0	50
g_2	0	0	1	25
g_3	0	1	1	10
g_4	0	1	0	15

Table 6.5: Errors Distributions

				Distribution A		*Distribution B*		*Distribution C*	
Granule	*a1*	*a2*	*a3*	*Sup(gi)*	*Distance*	*Sup(gi)*	*Distance*	*Sup(gi)*	*Distance*
g_2	0	0	1	15	7.86	3	-4.14	7	-0.14
g_3	0	1	1	5	-2.14	14	6.86	7	-0.14
g_4	0	1	0	6	-1.14	9	1.86	7	-0.14
g_5	1	1	1	4	-3.14	12	4.86	7	-0.14
g_6	1	0	0	3	-4.14	4	-3.14	7	-0.14
g_7	1	1	0	10	2.86	6	-1.14	7	-0.14
g_8	1	0	1	7	-0.14	2	-5.14	8	0.86

or patterns than DT_{A2}. This indicates that DT_{A1} presents more random errors than DT_{A2}. Based on the definition of randomness degree presented in paper Alkharboush and Li [2010], users can calculate the randomness degree RD_A of errant data as follows:

Definition 11. ***(Randomness Degree)*** *Let $|DT_A|$ is the number of errant granules g_i in DT_A and $2^{|Ai|}-1$ is the size of covering set for errant granules in DT_A. The random degree is defined as:*

$$RD_A = \frac{|DT_A|}{2^{|Ai|}-1} \tag{6.1}$$

The size of the covering set includes normal patterns (all A_i fields are 0) and errant patterns. In our example, the size of the covering set for DT_{A1} in Table 6.3 is $2^{|3|} = 8$. This covers one normal pattern and all potentially errant patterns generated from the database regardless of its number of rows. The size of the

covering set for DT_{A2} Table 6.4 is the same as Table 6.3 because both tables have the same attributes $\mid A_i \mid = 3$. Hence, Hence, users need to exclude normal patterns when they calculate randomness degree RD_A.

Referring back to the DT_{A1} in Table 6.3 and DT_{A2} in Table 6.4, we can apply Equation (6.1) to measure the RD_A of each. For example, DT_{A1} shows seven errant patterns, e.g., g_2 to g_8. By using the Equation (6.1), the RD_A of error in DT_{A1} is $7/7 = 100\%$. This means that error values cover all possible errant patterns in database T. However, in some scenarios, the number of errant patterns g_i in a decision table is less than the covering set size, as in DT_{A2}. In this case, we have low RD_A $3/7 \approx 43\%$.

6.5.2 Distinguish Between Systematic and Random Patterns

This section attempts to discriminate between systematic and random patterns. In the previous section. Equation (6.1) enabled user to calculate the randomness degree; and if $RD_A >$ threshold, then errors in T are considered to have a random distribution. For example, if the threshold for the decision tables shown in Table 6.3 and Table 6.4 is 50%, then the DT_{A1} in Table 6.3 has random errors with randomness degree 100%, and Table 6.4 has systematic errors. However, when we analyse DT_{A1} 6.3, it is noticeable that some errors occur more frequently in some patterns than in other patterns. This leads us to a critical question: is measuring randomness degree enough for a complete quality assessment solution?

To obtain deep insight into this problem, we examine three different decision tables with different distributions of errors see Table 6.5. Remember that we only

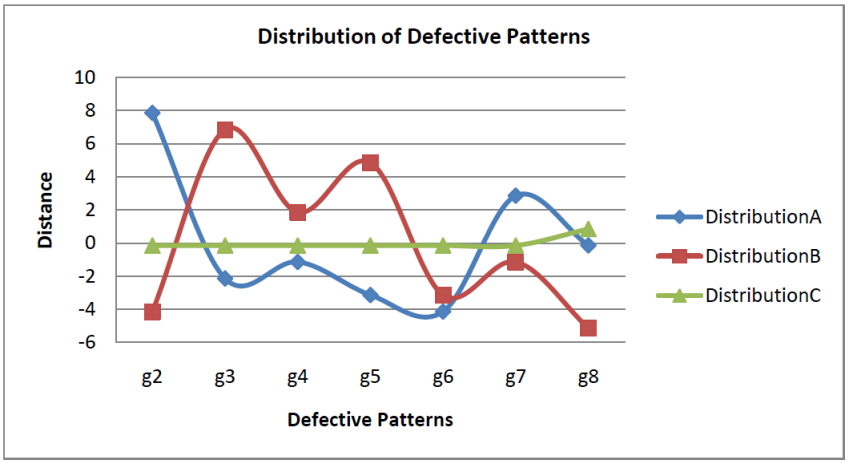

Figure 6.2: Distance Distribution

change the support on defective patterns to have different distribution weights, while the RD_A degree remains the same at 100%. Although all three distributions A, B and C in Table 6.5 have the same RD_A with randomness degree 100%, handling errors in a distribution like A is less complicated than in other distributions such as distribution C. Also, fixing several patterns e.g. (g_2 and g_7) in A could significantly improve the quality of a database. Hence, we enhance the RD_A presented in this chapter by introducing two measurements to categorise systematic patterns and random patterns into two groups. Also, we calculate the impact of each pattern on the quality. This enables decision makers to target the most serious errant patterns based on the time and resources available.

The first measurement is based on pattern support. In a decision table, each pattern has a support which indicates the frequency with which this type of error occurs in the database. Using this, users can specify a threshold to find the severity of defective patterns. If $Sup(gi) >$ threshold, the nominated pattern is considered to be a systematic pattern. For example, if users specify 8 as the minimum threshold for distribution A in Table 6.5, then we have two severe errant

patterns, g_2 and g_7. This solution can also be applied to distributions B and C but users need to modify the minimum threshold accordingly.

The second measurement for distinguishing between systematic and random patterns is based on the distance between patterns. For this, we measure the distance between defective patterns to find the location of systematic patterns S and random patterns R. More importantly, we distinguish between pure random patterns PR and weak random patterns WR.

Definition 12. ***(Define Systematic and Random Patterns)*** *Given a granule distance dis_{g_i} and the minimum distance min_dis_1, the systematic granule (S) and Random granule (R) are defined as:*

$$g_i = \begin{cases} S \quad if \quad dis_{g_i} > min\_dis_1 \\ R \quad = \begin{cases} PR \quad if \quad dis_{g_i} > min\_dis_2 \\ WR \quad if \quad dis_{g_i} < min\_dis_2 \end{cases} \end{cases}$$

To measure the distance, the user firstly needs to calculate the average support avg_sup of the defective patterns as in Equation 6.2. Then the user calculates the distance value dis_{g_i} between a patterns $Sup(gi)$ and avg_sup.

$$avg\_sup = \frac{1}{|DT_A|} \sum_{g \in DT_A} Sup(g_i) \tag{6.2}$$

We determine the severity of systematic or random errors based on the number of defective patterns in each category and their supports. For example, Figure 6.2 depicts the distance distributions for A, B and C. From this graph, we can easily see which pattern is far from the average support avg_sup and which is not, as presented in Table 6.5. However, Distribution C has the linear distribution of

Algorithm: Errant Granule

Input : T - a Table,
$A_i = \{a_1, a_2, ..., a_n\}$ // a Set of attributes
The minimum distance min_dis_1, min_dis_2

Output: Errant granules and the judgement about systematic and random

Step 1: // Get the decision table DT_A
Select $*$, count $(*)$ as Sup
From T
Group by a_1,a_2,...,a_n;
Remove the normal granule from DT_A

Step 2: // Calculate the Randomness Degree of Error Data
$RD_A = \frac{|DT_A|}{2^{|A_i|}-1}$;
//where $|DT_A|$ is the number of errant granules in DT_A
//and $2^{|Ai|}-1$ is the number of possible errant granules.

Step 3: // Calculate the distance for granules
$avg\_sup = \frac{1}{|DT_A|} \sum_{g \in DT_A} Sup(g_i)$;
foreach $g_i \in DT_A$ do
$dis_{g_i} = Sup(g_i)$ - avg_sup;
endfor

Step 4: // Decide systematic and random patterns
foreach $g_i \in DT_A$ do
if $dis_{g_i} > min\_dis_1$ then
g_i is a systematic pattern;
if $dis_{g_i} > min\_dis_2$ then
g_i is a pure random pattern;
else
g_i is a weak random pattern //or outlier pattern;
endif
endif
endfor

errant patterns. This indicates that these errant patterns mostly have random errors.

This study calls (systematic S and pure random patterns PR) as useful patterns (U) for solving potential quality problems. Also, we consider (weak random errant patterns WR) as outlier patterns and labelled as less useful patterns for quality problems. However, we highlight and measure the impact of both categories, useful (U) and less useful errant patterns (L), on quality improvement. Equation (6.3) calculates the quality improvement that user could obtain from correcting only the useful patterns.

$$quality\_improvement = \frac{\sum_{g \in U} Sup(g)}{\sum_{g \in DT_A} Sup(g)} \tag{6.3}$$

where $\sum_{g \in U} Sup(g)$ is the total support of useful patterns and $\sum_{g \in DT_A} Sup(g)$ is the total support of all defective patterns. This equation can also be used to calculate the the quality improvement for less useful patterns.

The above Errant Granule algorithm describes the procedure steps for finding the errant granules. The four steps cover constructing DT_A, calculating randomness degree and distance, and determining errors as either systematic or random.

6.5.3 Pattern Reduction for Random Patterns

In the above sections, we have answered two critical questions for data quality: how to calculate randomness degree and how to define systematic and random errant patterns. However, users are likely to have large numbers of defective weak random patterns WR. Things get even worse when the defective weak random

patterns have the same probability distribution as distribution C, shown in Table 6.5, with low frequency. In this case, all patterns have the same possibility and hence we cannot determine which patterns are going to be representative or have potential for solving a large percentage of quality problems, compared to the ones that have less impact on quality. Furthermore, reporting large defective random patterns to decision makers is not a practical solution. Therefore, we need to efficiently and effectively extract pure random PR error from weak random WR error in such a way that users do not lose knowledge nor deal with large and low frequency patterns.

For this reason, this study introduces a granule taxonomy structure to extract systematic and pure random errant patterns from weak random errant patterns. We start this granule taxonomy by vertically segregating the decision table DT_A, which has long patterns, into two categories based on attribute error rate. We group attributes with a low error rate into one category called the condition table (C) and group the remaining attributes which have a high error rate in another category called the decision table (D). Note that, $C \cap D = \emptyset$ and $C \cup D \subseteq A_i$. The details of how to construct condition table (C) and decision table (D) and how to generate the rules between these tables were discussed in Section 2.4.3.

Constructing condition table (C) and decision table(D) for small patterns enables user to see whether the sub patterns in (C) and (D) can provide significant and valuable information. After that, users can repeat our algorithm for each small granule to define systematic and pure random patterns from both condition and decision tables and use them as useful patterns for determining severe quality problems.

These steps can be repeated for further lower levels of decision table if the

errant patterns obtained are large and random. This solution enables users to significantly reduce the number of errant random patterns and extract a few useful potential patterns from them for solving potential quality problems.

6.6 Chapter Summary

Data quality assessment is a critical component for organisational performance. It supports decision makers to make correct decisions that meet organisational needs. Quality assessment has great benefit for consumer satisfaction, employee performance and operational costs. Most current approaches depend on error rate or accuracy rate to present the defective or correct ratio in a database. However, error rate or accuracy rate does not provide management with an accurate and reliable assessment as the errors can have different distributions over time. The decision rule method proposed in this thesis provides management with a reliable and efficient data quality assessment. It enhances decision makers' ability to clearly assess quality change in organisational information systems. By adopting a decision rule method, an organisation can examine whether the quality of data has improved or deteriorated. Managers and executives can rely on the decision rule method to estimate the time and costs required for conducting a quality improvement task.

This chapter also recognises that errors can systematically occur in specific locations or randomly appear across columns and rows. The proposed method provides users with an accurate assessment of the degree of randomness. This randomness measurement enables users to distinguish between systematic and random errors. This could have significant advantages as the users can identify

the location of the most severe errors and measure their severity. This benefits decision makers by allowing them to determine the techniques and resources needed to conduct quality improvement tasks.

Chapter 7

Experiments and Results

This chapter demonstrates the capabilities of the proposed solutions for detecting and assessing outlier data in a very efficient and effective way. Compared to several state-of-the-art techniques, tested with substantial experiments, the proposed algorithms show promising solutions for the automation of data quality assessment for outlier data. All the proposed algorithms are implemented in the *C++* language.

Table 7.1: Description of Real and Synthetic Data Sets

Data Set Name	*#Records*	*#Attributes*	*#Continuous*	*#Discrete*
Post-operative	90	9	0	9
Breast Cancer Wisconsin (Original)	699	9	0	9
Adult	45221	13	6	7
Syn.5000	5000	10	5	5
Syn.10000	10000	10	5	5
Syn100000	100000	10	5	5

7.1 Experimental Datasets

7.1.1 Synthetic Datasets

There are limited real datasets with labelled outliers that are publically available for experimentations. Therefore we generated four mixed-attribute synthetic datasets of different sizes using http://www.datasetgenerator.com. and generated the data as in Otey et al. [2006]. For each dataset, there are 5 continuous attributes and 5 categorical attributes and we specified the maximum distinct values in each attribute as 100 distinct values for datasets size 5000, 10000 and 100000(A). Then, we increased the number of maximum distinct values to 500 for each attribute in dataset 100000(B) in order to validate the proposed solution for a sparser dataset. For the experiment focusing on outlier detection, we generated the above datasets using different data distributions. Firstly, we used a normal distribution to generate the normal data points. Then we generated another small dataset that was uniformly distributed. The number of outlier points for the datasets were 5000, 10000 and 100000 (A and B) are 50, 100 and 1000 respectively. Figures 7.1 and 7.2 are examples of a synthetic data set and a real data set for outlier experiments.

Since the quality assessment methods apply to the binary matrix, we synthetically designed numbers of datasets with different noise rates. The values in these datasets were in binary format (0,1) where normal values are represented by 0 and noisy values represented by 1. We then created multiple datasets by gradually and randomly creating many binary matrices with different noise distributions. This enabled proper validation of the quality assessment and randomness degree.

ID	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
1	12	30	2	64	54	al	b	l	ah	e
2	2	15	6	13	40	g	g	p	au	k
3	16	30	33	71	1	aa	e	n	bh	g
4	18	2	42	19	55	l	a	f	ah	l
5	15	30	15	50	1	t	a	k	bc	h
6	17	23	1	2	1	n	d	m	l	s
7	15	32	37	21	54	ac	b	d	e	e
8	2	18	6	9	47	h	b	j	h	g
9	16	31	12	69	2	f	f	l	o	l
10	18	3	54	6	55	aa	g	f	n	v
11	11	30	52	13	13	h	g	s	aa	g
12	18	3	3	3	2	z	a	p	bc	j
13	14	30	53	53	54	ah	b	m	bi	a
14	3	26	49	14	33	n	a	q	ax	g
15	17	31	44	70	2	b	f	e	w	b
16	18	3	12	34	54	ae	a	d	bw	j
17	5	31	16	36	47	v	a	b	aa	b
18	17	31	3	2	1	p	d	i	aj	s
19	3	32	28	12	54	r	a	s	ah	w
20	1	13	34	62	30	p	d	p	ao	b

Figure 7.1: A sample of Synthetic Dataset

7.1.2 Real Datasets

During the research process we also validated the outlier algorithm using several real data sets that are available for download from UCI Machine Learning Repository http://http://archive.ics.uci.edu/ml/. Table 7.1 provides a description of each data set with regards to the number of records and attributes as well as the type of attribute; whether it is continuous or discrete. In the datasets 7.1, we tried to measure the effectiveness of the proposed outlier algorithms, not only those with different data sizes but also those with the increasing data dimensionalities. However, there is no definition of the outlier points in these datasets. Hence, to identify them we followed the method presented in Aggarwal and Yu [2001]; Hawkins et al. [2002]; Lazarevic and Kumar [2005], which relies on randomly reducing the class distribution for minor classes. The small class distribution and its correspondent features are considered outliers.

The Post-operative dataset determines whether or not patients should be admitted to the intensive care unit, be discharged, or rest in the hospital. The rare classes in this dataset are the first two, with labels I and S. These include a total of 26 points which are considered as outliers. The majority of the patients, with label A, are considered normal, and are given a total of 64 points. Although the size of Post-operative dataset is relatively small, it has been used by many authors for evaluating the effectiveness of their outlier algorithms.

The Breast Cancer Wisconsin dataset consists of 9 attributes and two other attributes named ID and class. The size of the data set is 699 records. There are two classes, labelled (2) and (4). The objects in class (2) are considered normal (or unmalignant). This class contains (458 or 65.5%). The remaining

ID	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15
1	48	Private	121424	Bachelors	13	Married-c	Exec-man	Husband	White	Male	0	0	55	United-St	>50K
2	56	Private	294209	HS-grad	9	Divorced	Prof-spec	Not-in-far	White	Female	0	0	48	United-St	<=50K
3	29	Private	285294	Assoc-acd	12	Never-ma	Adm-cleri	Unmarrie	Black	Female	0	0	40	United-St	<=50K
4	29	Private	168221	HS-grad	9	Married-c	Machine-(	Husband	White	Male	0	0	40	United-St	<=50K
5	57	Private	199847	Bachelors	13	Married-c	Exec-man	Husband	White	Male	15024	0	60	United-St	>50K
6	33	Private	117963	HS-grad	9	Never-ma	Machine-(	Other-rel	White	Male	0	0	55	United-St	<=50K
7	33	Private	181091	12th	8	Married-c	Craft-repa	Husband	White	Male	0	0	16	United-St	<=50K
8	44	Self-emp-	53956	Some-coll	10	Married-c	Machine-(	Husband	White	Male	0	0	57	United-St	<=50K
9	90	Self-emp-	83601	Prof-scho	15	Widowed	Prof-spec	Not-in-far	White	Male	1086	0	60	United-St	<=50K
10	26	Self-emp-	201579	5th-6th	3	Never-ma	Prof-spec	Unmarrie	White	Male	0	0	14	Mexico	<=50K
11	43	Private	104660	Masters	14	Widowed	Exec-man	Unmarrie	White	Male	4934	0	40	United-St	>50K
12	33	Private	227325	Assoc-acd	12	Never-ma	Other-ser	Not-in-far	White	Male	0	0	60	Scotland	<=50K
13	28	Private	129814	Some-coll	10	Separated	Craft-repa	Unmarrie	White	Male	0	0	50	United-St	<=50K
14	45	Private	341762	Masters	14	Married-c	Exec-man	Husband	White	Male	0	0	65	United-St	>50K
15	33	Private	204557	HS-grad	9	Married-c	Craft-repa	Husband	White	Male	0	0	40	United-St	<=50K
16	45	Private	155659	HS-grad	9	Married-c	Craft-repa	Husband	White	Male	0	0	40	United-St	>50K
17	34	Self-emp-	48935	Some-coll	10	Married-c	Farming-f	Wife	White	Female	0	0	30	United-St	<=50K
18	18	Private	93985	Some-coll	10	Never-ma	Other-ser	Own-chil	White	Male	0	0	30	United-St	<=50K
19	65	Local-gov	172646	9th	5	Married-c	Exec-man	Husband	White	Male	0	0	40	United-St	<=50K
20	52	Private	145409	10th	6	Married-c	Machine-(	Husband	White	Male	0	0	48	United-St	<=50K

Figure 7.2: A sample of Adult Dataset

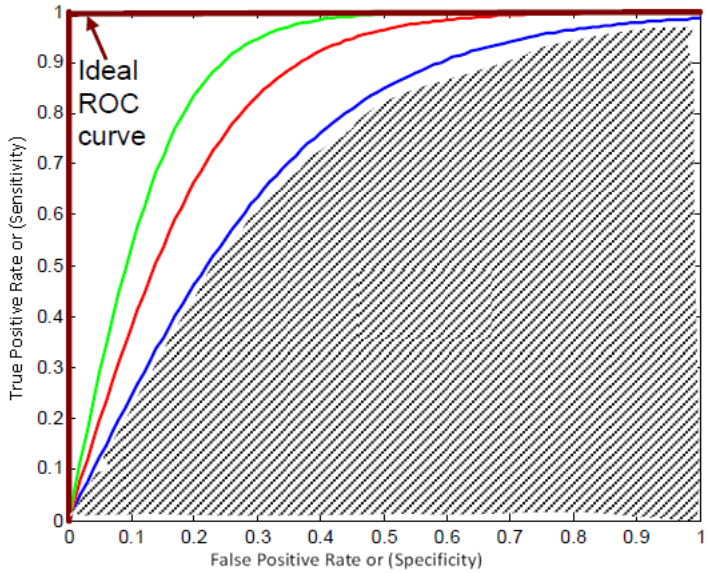

Figure 7.3: The ROC Curves for different detection algorithms

(241 or 34.5%)records that are assigned to class (4) are considered outliers (or malignant). We reduced the size of the outlier class (4) by 82% resulting in only 39 outliers. Then we randomly shuffle the order of the object instances. We also remove the 14 objects that contain missing values from the dataset. The modified breast cancer dataset includes 497 records which consist of 444 normal objects and 39 outliers.

The Adult dataset has two classes based on income: $\leq 50K$ and $> 50K$. The distribution of the classes is 76% and 24% respectively. We follow the same procedure as with the Breast Cancer Wisconsin dataset in order to make the class data more imbalanced by keeping only 800 points of the $> 50K$ class and considering these 800 points as outlier points in the adult dataset.

7.2 Performance Measurement

Outlier detection algorithms are usually evaluated by computing the ROC curve. The trade-off between the False Positive Rate (or Specificity) and the True Positive Rate (or Sensitivity) is illustrated in Figure 7.3. The ideal performance on the ROC curve occurs when the False Positive Rate is 0% and the True Positive Rate is 100%. Yet it is hard to achieve the ideal ROC curve particularly for real datasets where there is no existing predefinition of outlier points. Hence, the ROC curve in a real dataset is usually lower than the ROC in a synthetic dataset as the outlier points in a real dataset were defined based on changing class distribution and not based on all attributes. Therefore, it is possible that there could be some points in a non-nominated outlier class that behave more malignantly than the points in the rare class.

Since the outlier points are predefined in all our synthetic and real datasets, we can easily compute the following four confusion matrices: true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN). We used the result derived from the confusion matrix to compute the True Positive rate and the False Positive rate based on the following definitions:

$$TruePositiveRate = \frac{TP}{TP + FN}$$

$$FalsePositiveRate = \frac{FP}{FP + TN}$$

The true Positive Rate (or Sensitivity or recall) indicates the rate of correct outliers found by the algorithm, and the False Positive Rate represents the rate of misclassified outliers.

7.3 Experimental Setting

To achieve a fair comparison, we evaluated the quality of the proposed outlier algorithms with a number of well-known algorithms. The baseline algorithms were designed to detect outliers with mixed outlier attributes.

The results of algorithms He et al. [2005, 2006]; Koufakou et al. [2007]; Otey et al. [2006] for Post-operative and Breast Cancer Wisconsin were reported in Koufakou et al. [2007]. We compared our outlier algorithms to these algorithms He et al. [2005, 2006]; Otey et al. [2006]] as they are state-of-the-art methods for detecting frequent items as well, as being suitable for detecting outliers in a categorical dataset.

Additionally, we extensively validated our approach against the Orca algorithm Bay and Schwabacher [2003], which is a state-of-the-art algorithm for distance-based on outlier detection. The Orca algorithm handles outliers numeric, categorical or mixed attribute dataset because of the uses of Euclidean and Hamming distance measures. We specify the number of neighbours for the Orca algorithm to its 10 nearest neighbours. We also studied the impact of increasing the number of the nearest neighbours on outlier detection rate.

7.4 Evaluation Process

To implement a proper validation of the proposed algorithms in this thesis, all the algorithms presented in this study and the algorithms for comparison have been implemented in the C++ language. All the proposed algorithms have been compared to a number of state-of-art algorithms to determine the effectiveness of the proposed algorithms. The following Figure 7.4 illustrates the evaluation process of the proposed automated data quality assessment for outlier data.

- Data Preprocessing: The major problem in conducting an outlier detection study is that there are no predefined datasets that describe how outlier points ought to be defined. Therefore, this study adopts two popular approaches to artificially designed outlier points. The first approach is to randomly change the class distribution; this technique is used in number of real supervised datasets. To generate an outlier class in these supervised datasets, the user needs to randomly reduce the minor class creating what is called a rare class. Records belonging to the rare class are labelled as outlier data. For the second technique involving designing outlier data, the user needs to artificially design two different synthetic datasets with different distributions. For normal data, the data are distributed to fit the normal distribution. The outlier data points are distributed based on the uniform distribution. Then these two synthetic datasets with normal and uniform distributions are randomly shuffled and grouped in a single dataset where the user predefines normal and outlier data.

- Mining Outliers: After the normal and outlier data are predefined and grouped in a targeted dataset, we run our algorithms and the baseline

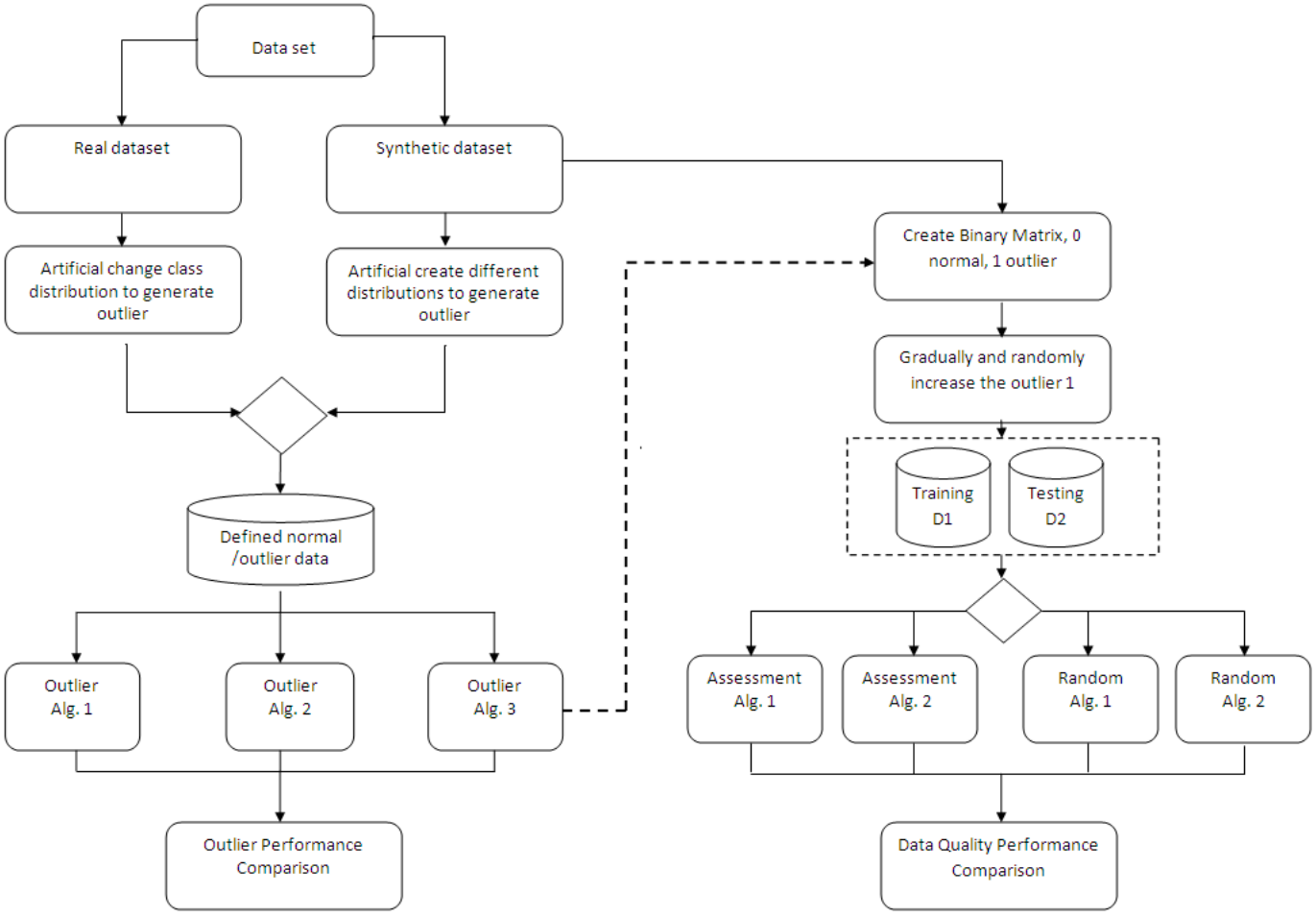

Figure 7.4: Evaluation Process

algorithms. This study firstly specifies the desired number of Top K outlier points. Then this study measures the accuracy of the returned points in detecting the outlier data from the original datasets.

- Performance Measurement: The effectiveness comparison between the proposed algorithms and the baseline algorithms is based on the ROC curve evaluation measurement. The computation of the ROC curve enables users to measure the trade-off between the False Positive Rate (or Specificity) and True Positive Rate (or Sensitivity).

- For evaluation of the randomness and assessment measurements proposed in this thesis, this study generates several binary matrix datasets where 1 represents normal data and 0 represents random outlier points. The outlier data are synthetically and randomly increased in order to effectively determine the capability of the proposed algorithms in assessing quality change and measuring the randomness degree of error data over time. The dashed line in Figure 7.4 can be used to derive the real binary matrix from the dataset. Since the goal of the proposed quality assessment is to assess any quality change with regard to outlier data. This study assumes that there is a Training (historical) dataset D1 and a Testing (newly generated) dataset D2.

- Quality Assessment: For assessing quality change, we run our decision rule algorithm on D1 and D2 and compare the results to the results found p-value to determine the effectiveness of the proposed algorithm in capturing quality changes in outlier data.

- Randomness Measurement: The user can utilise the randomness degree for outlier data in D1 and compare it to D1. This study evaluates the outlier data from several synthetic datasets and compares it to the well known LZ randomness algorithm. Since the quality assessment and randomness measurement are not as complicated as the outlier study, we directly use a simple rate to compare quality assessment and randomness degree algorithms.

7.5 Experimental Results for Outlier Algorithms

7.5.1 GBOD Algorithm for Outlier Detection

The experimental results in this section prove that the use of the weighted discernibility matrix proposed in this thesis provides accurate results that can be employed for capturing outlier values. Table 7.2 compares the GBOD results to a number of state-of-the-art algorithms for the Breast Cancer Wisconsin dataset. For example, when the number of top records was 4, we found that the returned 4 records from GBOD and Greedy and AVF are outliers with a 0% False Positive Rate; whereas the other baselines such as Orca, FPOF and Otey's return 3 out of 4 outlier records and 1 record was falsely predicted as an outlier.

As can be seen from Table 7.2 GBOD delivers a higher accuracy for all the different numbers of Top N outlier points. Particularly for the Top 24, 32 and 40, GBOD returns more accurate outliers, whereas the other baseline algorithms have incorrectly flagged many normal points as outliers. From Table 7.2, it is clear that the accuracy of GBOD for detecting outliers is stable with increasing numbers

Table 7.2: Breast Cancer Wisconsin dataset

# of record	*GBOD*	*Orca*	*Greedy*	*AVF*	*FPOF*	*Otey's*
4	*4*	3	4	4	3	3
8	*8*	7	8	7	7	7
16	*15*	15	15	14	14	15
24	*23*	20	22	21	21	21
32	*31*	26	29	28	27	28
40	*36*	33	33	32	31	33
48	*38*	37	37	36	35	37
56	*39*	39	39	39	39	39

Table 7.3: Post-operative dataset

# of record	*GBOD*	*Orca*	*Greedy*	*AVF*	*FPOF*	*Otey's*
10	*5*	6	4	3	3	1
20	*9*	10	7	7	7	7
30	*14*	13	8	10	9	9
40	*19*	17	12	11	10	10
50	*23*	21	13	12	12	13
60	*24*	23	20	16	17	18
70	*26*	25	21	21	21	21
80	*26*	26	24	24	24	24
90	*26*	26	26	26	26	26

of Top N outliers. Unlike GBOD, the accuracy of the comparison algorithms fluctuates with increasing number of Top N outliers.

Table 7.3 illustrates the comparison results for another real dataset called the Post-operative dataset. Although the Post-operative dataset is very small, GBOD still provides accurate outlier detection results. From Table 7.3, it is clear that GBOD outperforms the following algorithms which rely on frequent item patterns: Greedy, AVF, FPOF, and Otey's, presented in He et al. [2005, 2006]; Koufakou et al. [2007]; Otey et al. [2006] respectively.

Comparing GBOD and Orca, the state-of-the-art distance-based method, GBOD returns more accurate outliers for numbers of Top N outliers between 30 and 90. When the number of top outliers returned was 10 and 20, Orca returned 6 and 10 accurate outliers respectively; whereas the number of accurate outlier returned by GBOD is slightly lower than Orca with 5 outliers out of top 10 and 9 outliers out of top 20. Table 7.3 shows that GBOD returns all the 26 outliers when the specified Top N was 70. For the remaining algorithms, all the 26 outliers were found in the Top 80 for the Orca algorithm and Top 90 for algorithms Greedy, AVF, FPOF, and Otey's.

Since the results of both the GBOD and Orca algorithms are to some degree similar to each other and outperform the Greedy, AVF, FPOF, and Otey's algorithms, this thesis includes several experimental studies to investigate the quality of both GBOD and Orca for more large datasets.

Firstly, GBOD is compared to Orca in the 5K synthetic dataset. The number of outliers in this dataset is 50. Table 7.4 shows the effectiveness of both GBOD and Orca for accurately defining outlier points. It is clear that GBOD still maintains higher accuracy for all Top N.

For further validation of the effectiveness of the proposed GBOD algorithm, we measured the quality of outliers found by GBOD in a larger dataset with 10000 records. The results for the GBOD algorithm for this dataset are outstanding, since none of the retrieved points are mislabelled, as shown in Table 7.5.

The results for GBOD in both real and synthetic datasets demonstrate the capabilities of the proposed GBOD algorithm in terms of providing effective outlier detection methods. The studies have proven that the new weight discernibility matrix described in Chapter 5 can be utilised to compute the distance between

Table 7.4: 5K Synthetic Dataset

Top N	***GBOD***	***Orca***
5	4	3
10	7	6
15	12	8
20	16	12
25	19	15
30	23	17
35	25	20
40	28	23
45	32	26
50	37	30

Table 7.5: 10K Synthetic Dataset

Top N	***GBOD***	***Orca***
10	10	8
20	20	17
30	30	26
40	40	35
50	50	44

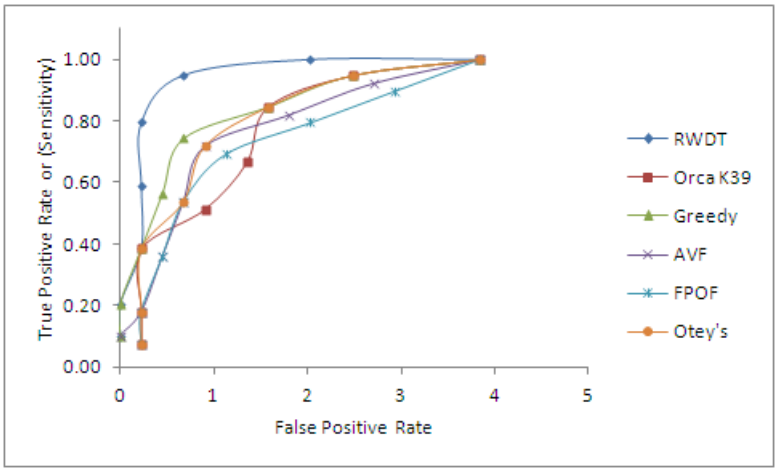

Figure 7.5: ROC for Breast Cancer Wisconsin Dataset

granules and points as accurately as using the traditional Euclidean distance.

7.5.2 RWDT Algorithm for Outlier Detection

7.5.2.1 Results and Discussions

This section shows the results for the RWDT algorithm and tests its effectiveness when used for capturing outlier data. Figure 7.5 illustrates the ROC curves of RWDT and the other baseline algorithms. As can be observed, RWDT has the highest ROC curves compared to the other algorithms.

The Figure 7.6 demonstrates the quality of RWDT compared to the other five baseline algorithms. From figure 7.6, it is clear that RWDT and Orca outperform other baselines. It is also noticeable that the results for Orca and RWDT are relatively similar; with Orca performing better than RWDT in the first three cut-offs. We anticipate these results because the size of the Post-operative dataset is very small.

Since the performance of the RWDT and Orca algorithms have results close

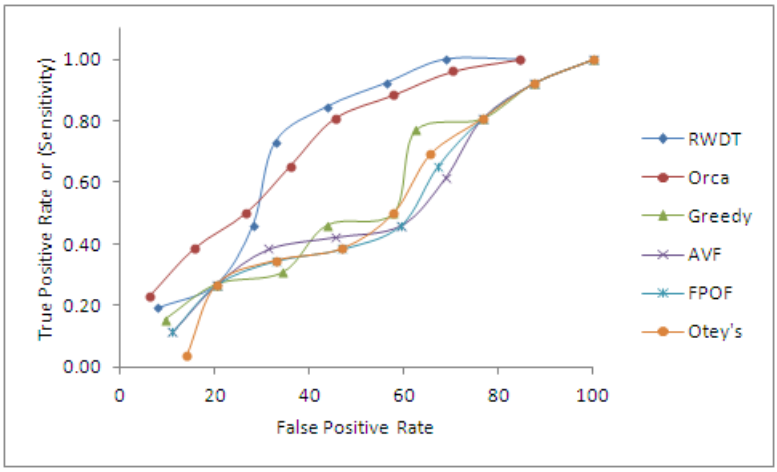

Figure 7.6: ROC for Post-operative Dataset

to the ROC curve, particularly in the Post-operative dataset, we extensively compare the proposed RWDT algorithm to Orca in a number of large dimensional synthetic and real datasets.

In Synthetic dataset-5000, we predefined 50 outlier points. As can be observed from Figure 7.7, all 50 outlier points were detected. Figure 7.7 also shows that the RWDT algorithm has a higher ROC curve than the baseline. Hence, ,the False Positive Rate for the RWDT algorithm is less than that for the Orca algorithm, which means that the more accurate outlier detection was carried out by the RWDT.

We also measured the quality of the proposed solution with an increased database size of 10000 points with 100 outlier points. As shown in Figure 7.8, the ROC curve for RWDT outperforms that for the Orca algorithm in all different axis scales.

In our experiment, we considered the possibility that the performance of algorithms might change from dense datasets to sparse datasets. Therefore we

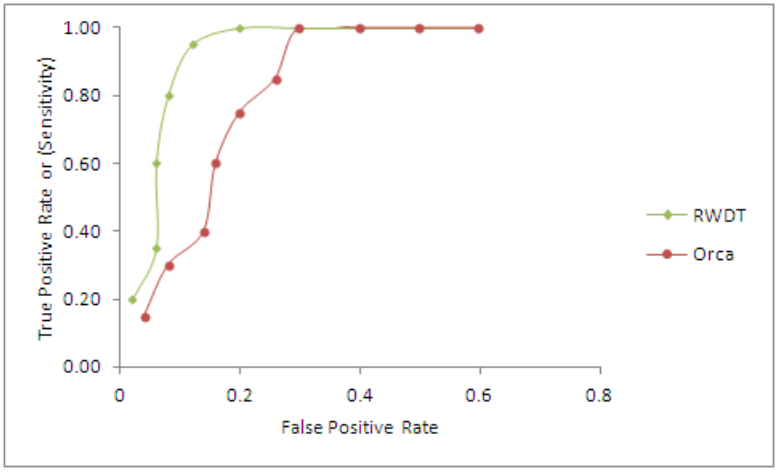

Figure 7.7: Synthetic dataset-5000

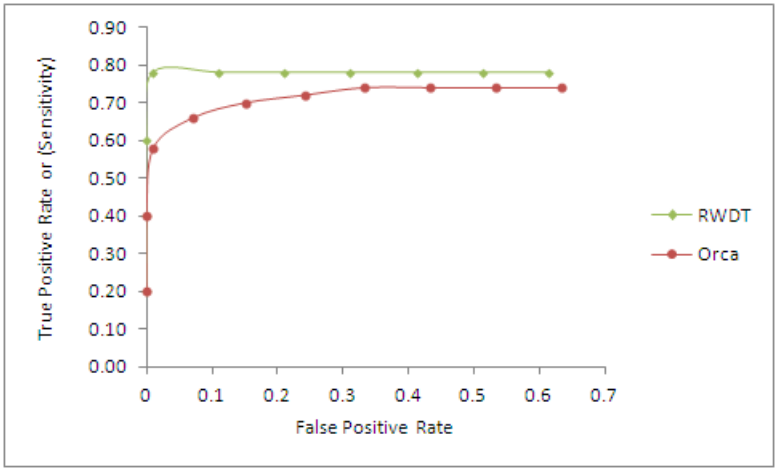

Figure 7.8: Synthetic dataset-10000

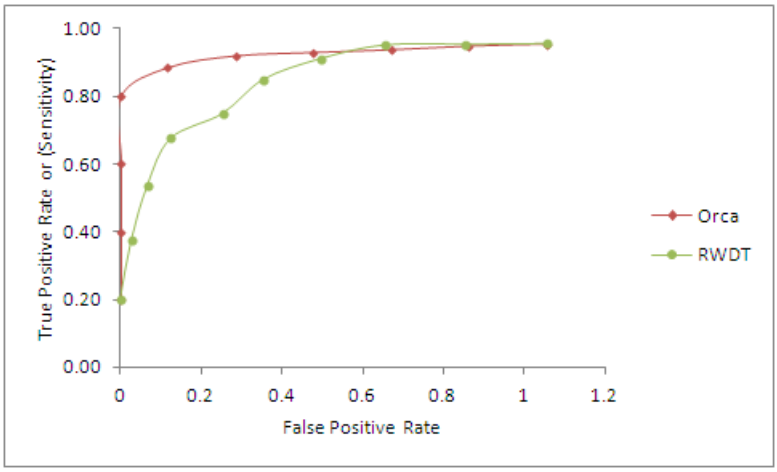

Figure 7.9: Synthetic dataset-100000(A)

generated two synthetic datasets of the same size with different densities, called dataset-100000(A) and dataset-100000(B). We specified the maximum number of distinct values for dataset 100000(A) to give 100 distinct values for each attribute. We created the data sparser in dataset-100000(B) by increasing the maximum number of distinct values to 500 for each attribute.

For Synthetic dataset-100000(A) with 1000 outlier points, it was clear from Figure 7.9 that the Orca algorithm has a higher ROC curve than RWDT. The reason is that the number of distinct values selected for the synthetic data set was small, with 100 distinct values for each attribute. Hence, when we repeat the test with a less dense dataset, such as dataset-100000(B), we notice that the RWDT produces a much higher ROC curve than the Orca algorithm, as shown in Figure 7.10. Even when we increase the number of nearest neighbours from 10 to 20, RWDT still has higher ROC results.

Additionally, we compared the RWDT and Orca algorithms to the following real datasets: Adult, Post-operative, and Breast Cancer Wisconsin. Figures

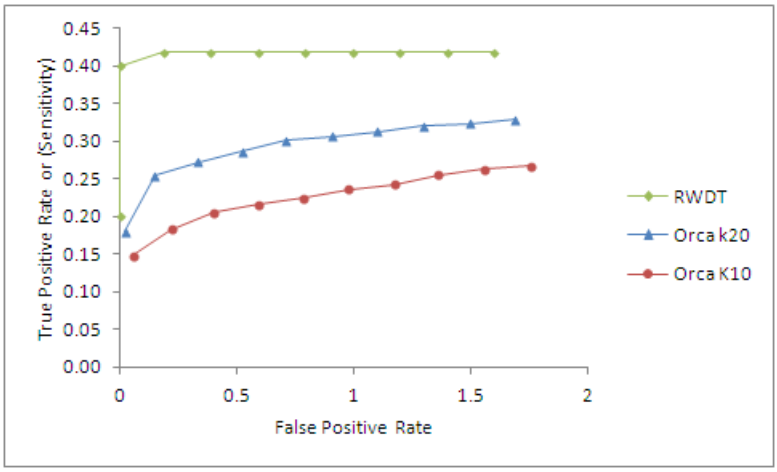

Figure 7.10: Synthetic dataset-100000(B)

7.11 and 7.12 show the ROC curves for the RWDT and the Orca algorithms for the Adult and Post-operative datasets. As Figures 7.11 and 7.12 show, RWDT outperforms Orca in both the Adult and Post-operative datasets respectively.

We also studied the change in outlier detection rate for different specified values of the nearest neighbour parameter for the Orca algorithm. Figure 7.13 compares the RWDT and Orca algorithms. As can be observed, the quality of outliers improves as we increase the number of nearest neighbours. However, when we specify the number of nearest neighbours for the Orca algorithm, so that it is the same as the number of outliers in Breast Cancer Wisconsin dataset, RWDT still obtains a higher ROC curve.

7.5.3 CG Algorithm for Outlier Detection

This algorithm attempts to overcome the limitations of the distance-based method, in particular the sensitivity to the specification of the number of nearest neighbours, as well as the specification of minimum distance.

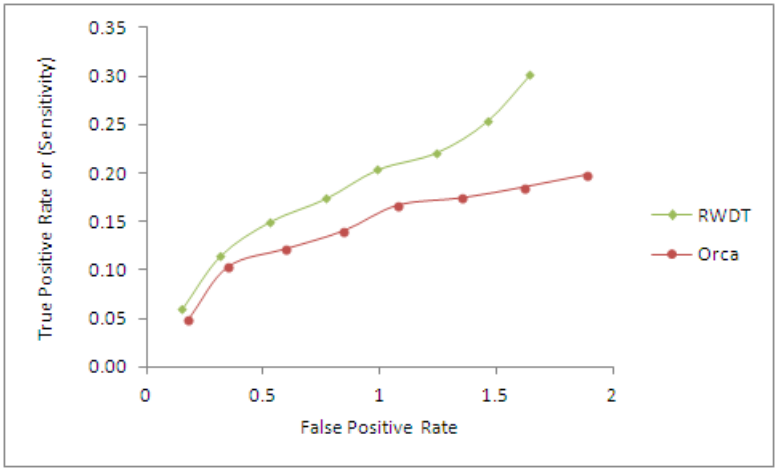

Figure 7.11: ROC for Adult Dataset

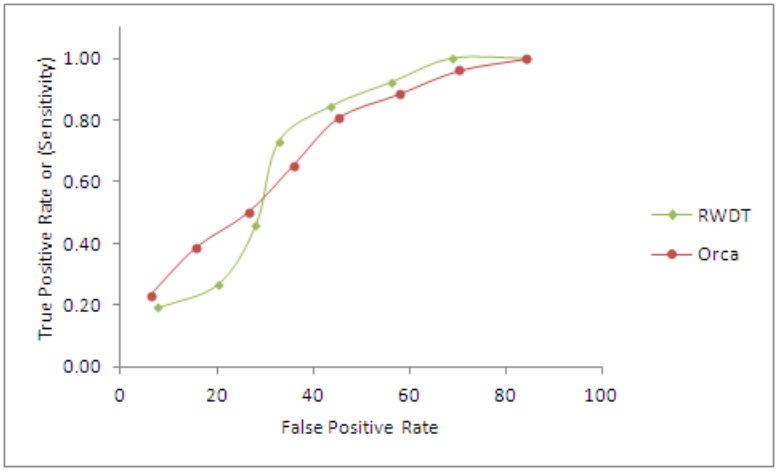

Figure 7.12: ROC for Post-operative Dataset

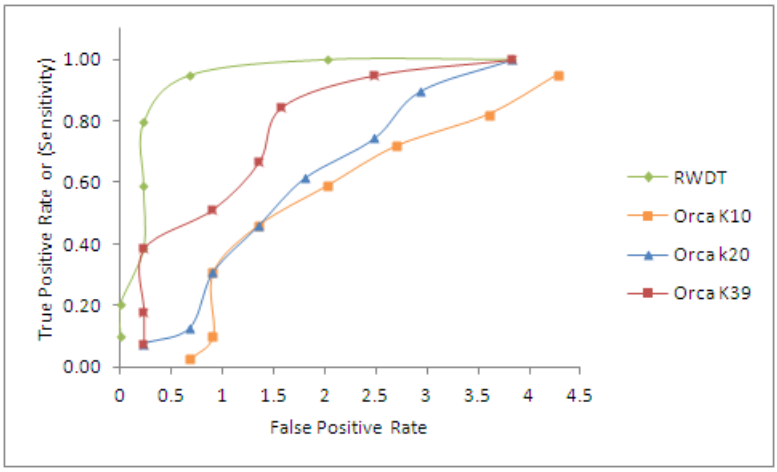

Figure 7.13: ROC for Breast Cancer Wisconsin dataset

The CG algorithm approaches this problem by efficiently and correctly finding the centroid granule CG in the dataset. This has great advantages in reducing the running time as the algorithm will compute the distance between the approximate L_G set and CG.

From the experiment, we can present two results for the CG algorithm. The first experiment, CG1, computes the centroid (average) from all granules G in WDT and the second, CG2, computes the centroid (average) granule from the H_G set only.

Figure 7.14 shows the ROC curves for CG1 and CG2 compared to the Orca algorithm, and it can be seen that CG1 and CG2 both perform better than Orca. Even after we increase the number of nearest neighbours from 10 to 39 points, the accuracy of CG1 and CG2 is higher than that of Orca. Furthermore, the CG algorithm proves its effectiveness for use with large datasets. Figures 7.15 and 7.16 demonstrate the improvements in the CG1 and CG2 results over those for the Orca algorithm.

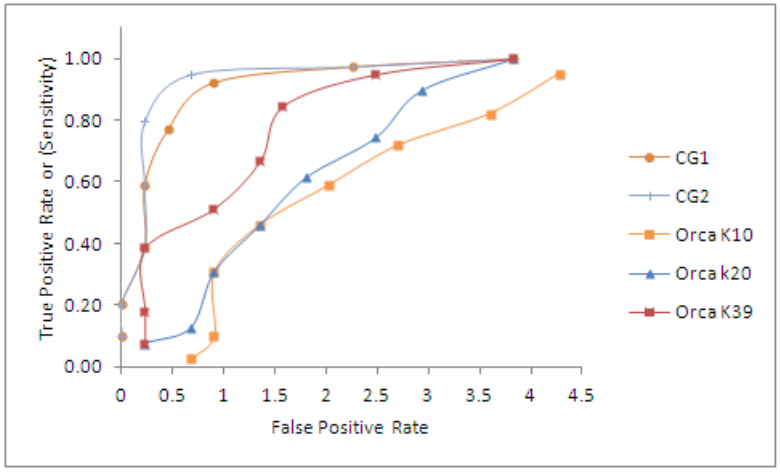

Figure 7.14: ROC for Breast Cancer Wisconsin dataset

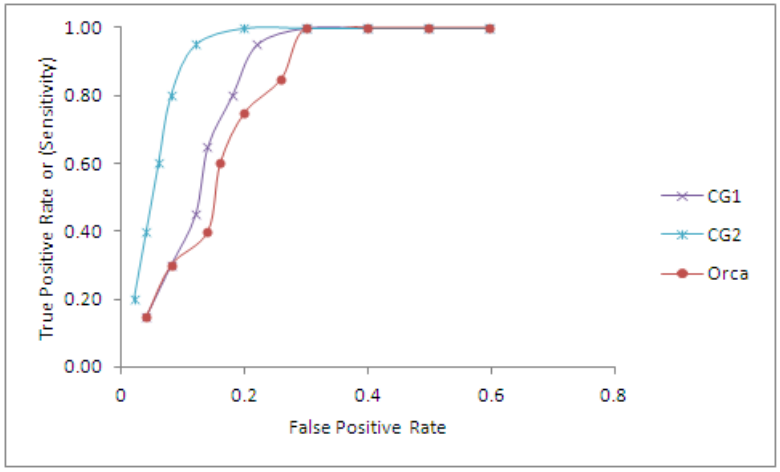

Figure 7.15: ROC for Synthetic dataset-5000

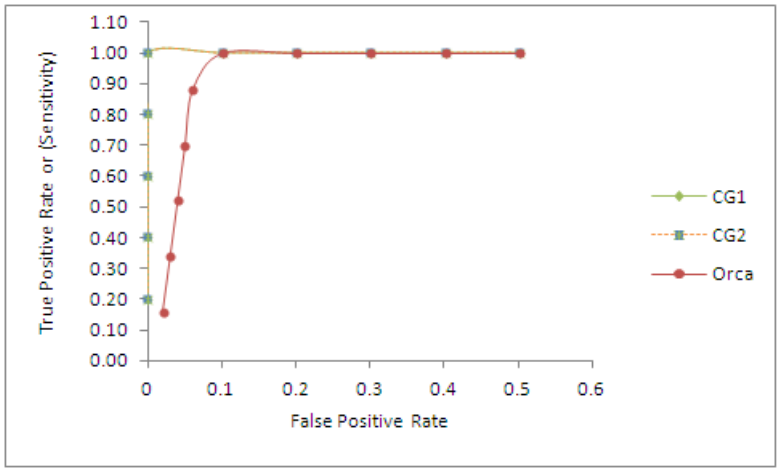

Figure 7.16: ROC for Synthetic dataset-10000

From the Figures CGforBreast, 7.15 and 7.16, we notice that the centroid found from H_G as CG2, provides more accurate results than the centroid found by all G, CG1. This proves that our approximation, which enables us to classify granules into H_G and L_G LG is accurate. Additionally, we can conclude that by computing the centroid from H_G, we reduce the impact of the other L_G granules on the results.

7.5.4 Comparison Between GBOD, RWDT and CG Algorithms

This section provides a sensitivity analysis for the proposed algorithms. Firstly, the intent is to study and compare the effectiveness of the proposed algorithms for the purpose of exposing outlier data. Then, the study will compare the size of candidate granules by applying each of the algorithms.

Table 7.6 shows a comparison of the results obtained by the proposed algorithms. As can be seen from Table 7.6, the Centroid Granule CG2, where the

Table 7.6: The Proposed Algorithms Results for synthetic dataset-5000

Top N	***GBOD***	***RWDT***	***CG1***	***CG2***
5	4	4	3	**4**
10	7	7	6	**8**
15	12	12	9	**12**
20	16	16	13	**16**
25	19	19	16	**19**
30	23	23	19	**24**
35	25	25	22	**27**
40	28	28	25	**31**
45	32	32	28	**35**
50	37	37	32	**39**

CG is computed from the HG, outperforms the other GBOD, RWDT, CG1 algorithms. This is particularly the case from the Top 30 and upward as the accuracy of the CG2 is higher than that for the other algorithms. For example, when the top N was 50, the CG2 returns 39 true outliers and 11 falsely predicted points. Whereas, the GBOD, RWDT returns the same result: 37 with 13 falsely predicted. The worst result for this data set was that given by the CG1 algorithm, with 32 true outlier points and 18 falsely predicted points.

The Figure 7.17 shows the precision of the proposed algorithms with different selected Top N.

Figure 7.18, illustrates the trade-off between the false and true positive rates for the proposed algorithms.

Additionally, the study examines the effectiveness of the results that were acquired using the proposed algorithms to evaluate the real Breast Cancer Wisconsin dataset. Figure 7.19 shows the results comparatively.

From examination of both the synthetic and real datasets in figures 7.18 and 7.19, the accuracy of the CG2 for detecting outlier data is shown to be higher

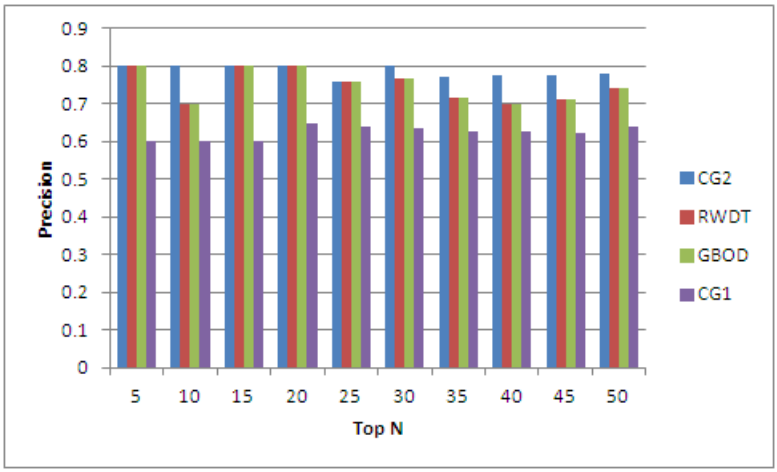

Figure 7.17: The Proposed Algorithms with Different Top N for Synthetic dataset-5000

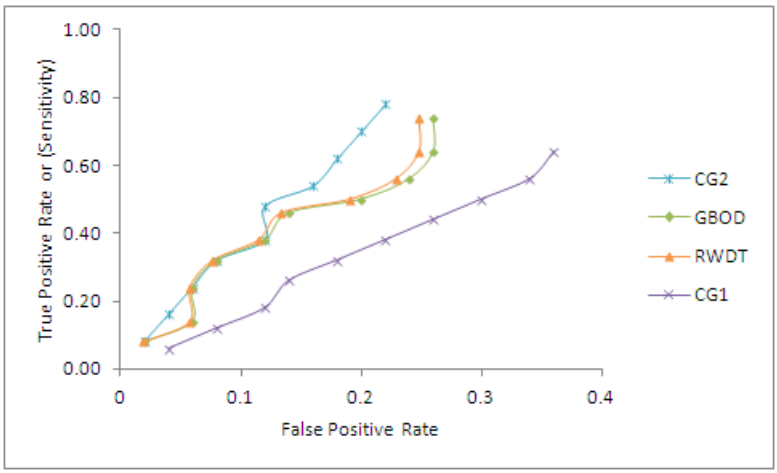

Figure 7.18: ROC for the Proposed Algorithms for Synthetic dataset-5000

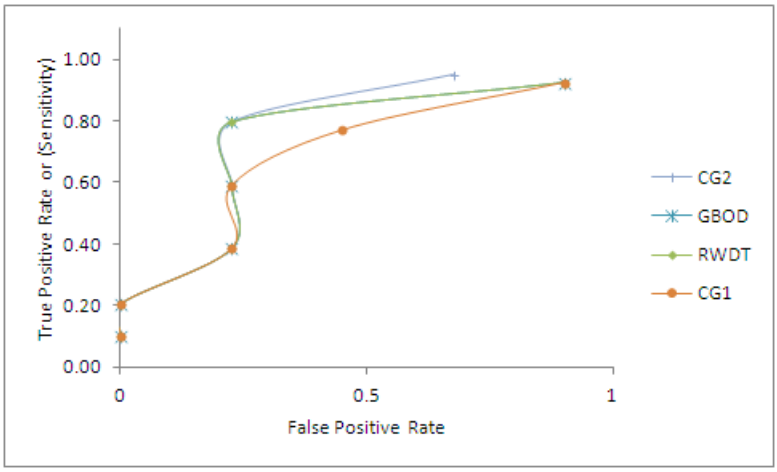

Figure 7.19: ROC for the Proposed Algorithms for Synthetic dataset-5000

than the algorithms: RWDT, CG1 and GBOD. However, the gap between the proposed algorithms reduces when these algorithms are applied to a real dataset, as shown in Figure 7.19.

The study also investigates the number of granules in L_G that are expected to hold outlier data. Since the traditional DT table relies on the degree of support to determine H_G and L_G granule as in GBOD algorithm, the size of the L_G set which is likely to hold outlier data is significantly large compared to the H_G set. This reduces the efficiency of the proposed GBOD when detecting outlier. This is particularly the case when the dataset has very few numeric attributes, because finding granularity in a dataset with numeric attributes can be difficult.

For example, Figure GBODdistrbution illustrates the granule distribution based on the support degree for the Adult dataset using GBOD algorithm. As can be seen in the Figure GBODdistrbution, the size of the L_G set is very large, covering 91.5% granules. Whereas, the size of the H_G is very small covering 8.5% of all granules.

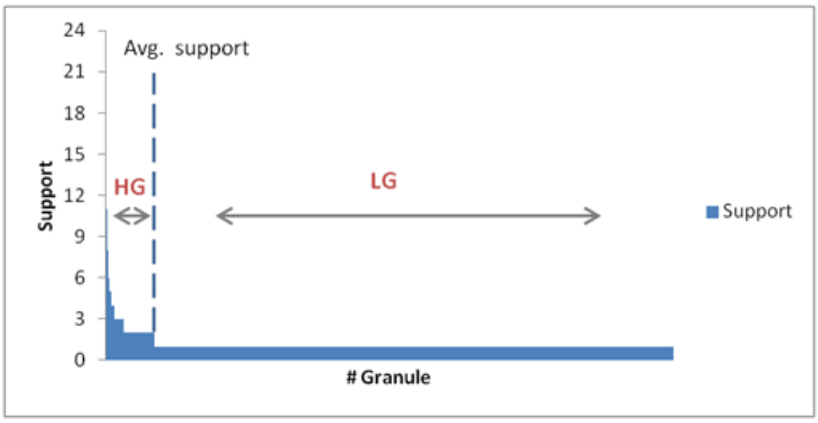

Figure 7.20: Distribution of the GBOD for Adult Dataset

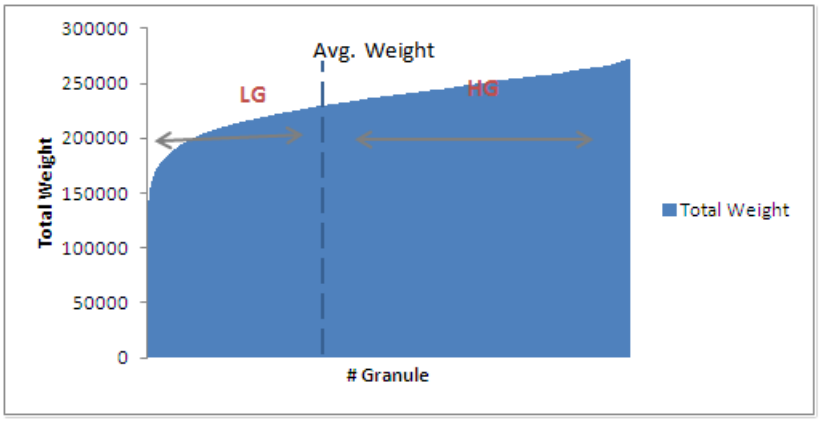

Figure 7.21: Distribution of the RWDT for Adult Dataset

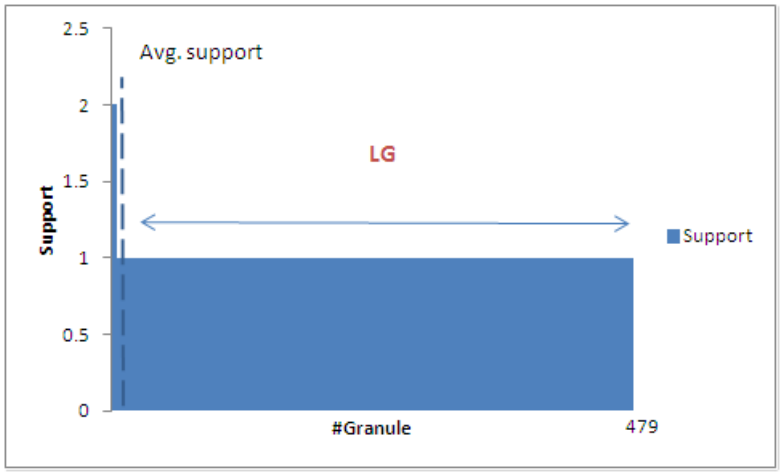

Figure 7.22: Distribution of the GBOD for Breast Cancer Wisconsin dataset

To overcome the limitations resulting from using the GBOD algorithm, and to reduce the mining space L_G set, this study introduces the RWDT algorithm. With the RWDT algorithm, the total weight of the granule is utilised to identify the H_G and L_G granules. For example, Figure RWDTdistrbution, illustrating RWDT distribution, shows the granules' distribution based on the total weight, as described by the RWDT algorithm for the Adult dataset. It is evident from the figure that the size of the L_G in RWDT is very small compared to that using GBOD, at 44.7%. This means that the RWDT algorithm mines just 44.7% of the data to find the outliers. Whereas, the GBOD algorithm mines 91.5% of the data for the same purpose.

As mentioned above, this study populates the distribution of another real dataset, called the Breast Cancer Wisconsin dataset to compare the differences between the L_G from the GBOD and the L_G LG from the RWDT, as described in Figures 7.22 and 7.23 respectively.

The number of candidate granules, L_G from the GBOD algorithm covers 99%

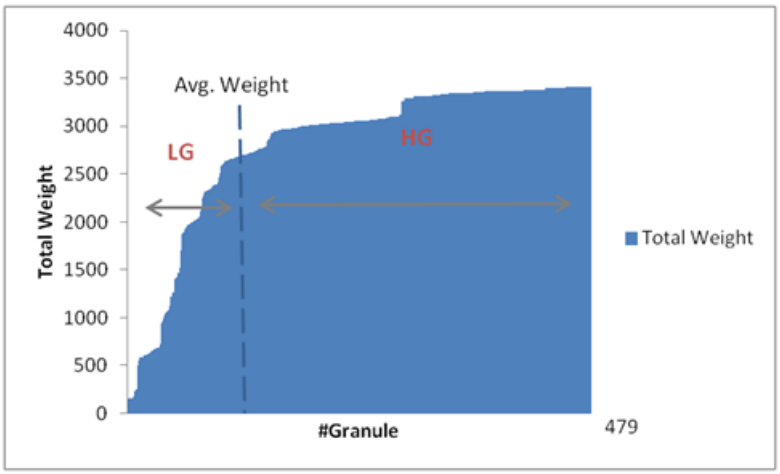

Figure 7.23: Distribution of the RWDT for Breast Cancer Wisconsin dataset

f the data as shown in Figure 7.22. Unlike the GBOD, the number of granules, L_G that are likely to hold outlier values based on RWDT is 29% of the original dataset size (see Figure 7.23). Mining such small candidate granules L_G from RWDT has great advantages as the mining space is much smaller than the mining space in the GBOD algorithm.

In the case of the CG algorithm, all the H_G granules are represented by a single granule. Instead of computing the distance from the L_G granule to number of H_G to determine the similarity or deviation degree, the CG algorithm only uses a single granule to represent all H_G granules. The CG algorithm computes the distance between L_G and the representative of H_G which is the CG granule.

Table 7.7: Decision Rules for D_1 and D_2

Database	$\boldsymbol{D_1}$	$\boldsymbol{D_2}$
Original	73	67
5%	277	286
10%	376	373
15%	495	472

7.6 Experiential Results for Quality Assessment

7.6.1 Decision Rule for Data Quality Assessment

This study syntactically generates four different distributions of binary datasets (0,1) with 10 attributes and 2977 rows. The distributions of errors in these datasets increases gradually by 5%,10% and 15% in order to examine the effectiveness of the decision rule method when assessing the quality change in datasets. The results obtained from the four databases (original, 5%, 10% and 15%) show encouraging results.

The assumption is in the decision rule method is that there are two datasets (D_1 and D_2) generated from different time period that have the same error rates degree. If the both D_1 and D_2 have the same errors locations then the rules of the D_1 match the Rules in D_2. TO comply with this assumption, the study randomly and equally divides each of four databases (original, 5%, 10% and 15%) into two part training set or D_1 which consists of 1489 rows and testing set D_2 which contains 1488 rows. D_1 is a history database and D_2 is a newly generated database, see Table 7.7.

7.6.1.1 Results and Discussions

In both D_1 and D_2, we construct decision tables for all four databases (original, 5%, 10% and 15%). This will help to group all similar rows together and measure the frequency of similar rows. After compressed a transaction records of each database, we obtain the decision table which includes numbers of rules or "granules " as in Table 7.7. For example the numbers of rules in the original database for D_1 which has 1489 rows is 73 rules and for D_2 which has 1488 rows gets 67 rules. Constructing a decision table significantly reduces the size of a database without loss of information occurring. The goal is to test whether the data tests with the same error rates have the same error patterns for D_1 and D_2 , in order to determine quality change. If the number of rules in D_1 comply with the number of rules in D_2, then the data quality for D_1 is the same as the data quality in D_2. If there are many new defective rules in D_2, then there is a quality problem, as new errant patterns appear. The support column in a decision table is also used to determine unmatched rules with either severe quality problems, or non-severe problems.

Decision rules can be used to estimate the probability of the new errant error patterns arising in the D_2. Errors are likely to appear across columns and rows with a degree of variance . Hence, the proposed decision rule method measures the probabilities of errant patterns arising, with regards to the impact on data quality, when dividing the support for each deficient rule as generated in D_2 to the total support for defective rules. This will determine if the newly generated rules in D2 incur critical quality problems.

Table 7.8 summarises the match and unmatched rules between the D_1 and

Table 7.8: Rate of Match and Unmatched Rules for D_1 and D_2

Database	$\boldsymbol{D_1}$	$\boldsymbol{D_2}$	***Rule Match(%)***	***Rule Unmatch (%)***
Original	73	67	77.1	22.9
5%	277	286	66.8	33.2
10%	376	373	68.4	31.6
15%	495	472	68.4	31.6

D_2 sets for all four databases. Although, these four data sets have the same error rates, the data quality in the D_1 and D_2 is different. The proposed decision rule provides accurate assessment of data quality. In Table 7.8, the rates for unmatched rules for the original database, the 5% database, the 10% database, and the 15% database are 22.9% , 33.2%, 31.6% and 31.6%, respectively. This indicates that there are new errant patterns appearing in D_2, alongside the increasing error rate.

We also examined the probability of unmatched rules that will determine the severity of the new errant patterns on the data quality. In this study, all four databases show no severe quality problems on unmatched rules because the probability of new errant rules in D_2 can be as small as 0.01% or maximum as 0.05%.

Since there are no existing methods with which to compare the proposed decision rule most quality assessments rely on error or accuracy rate. This study compares the proposed method to the t-test. The p-value from the t-test determines a significant divergence in the populations D_1 and D_2.

To compute the t-test, first, we calculate the error rate for each column in D_1 and D_2 for all the following databases: the original database, the 5% database, the 10% database, and the 15% database. Then, we calculate t-test to compare

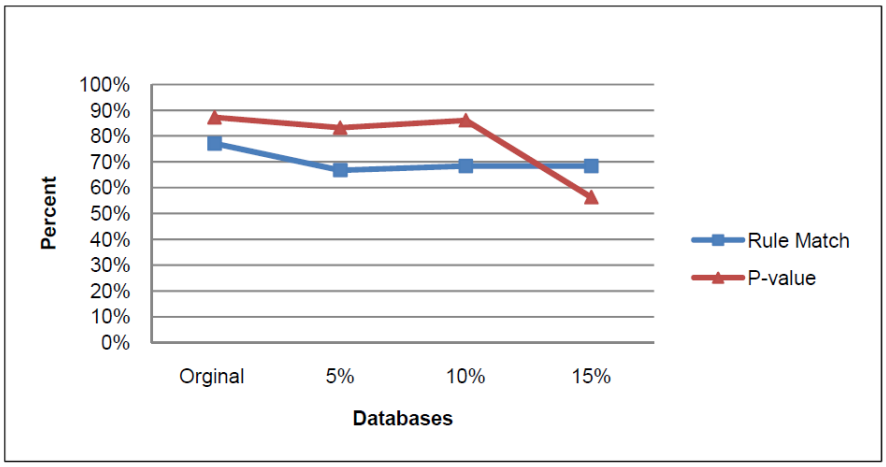

Figure 7.24: Compare Decision Rule with P-value

Table 7.9: Compare the Rate of Matched rules with P-value

Database	***Rule Match(%)***	***P-value (%)***
Original	77.1	87.2
5%	66.8	83.2
10%	68.4	86.1
15%	68.4	56.3

the error rate in D_1 with error rate in D_2.

The results of the p-value for these databases are then compared with corresponding decision rules in Table 7.9. In Figure 7.26, the decision rule method is shown to have a similar attitude to the p-value in the databases describes as original 5% and 10%. In the 15% database the p-value seem to be impacted by the increasing numbers of defective values. Therefore, in the 15%, database, the p-value displays a different attitude from the decision rule.

The results obtained from this study are encouraging and prove the effectiveness of the proposed decision rule method when assessing the quality of a large sized database. Unlike other studies which rely on error rate, the decision

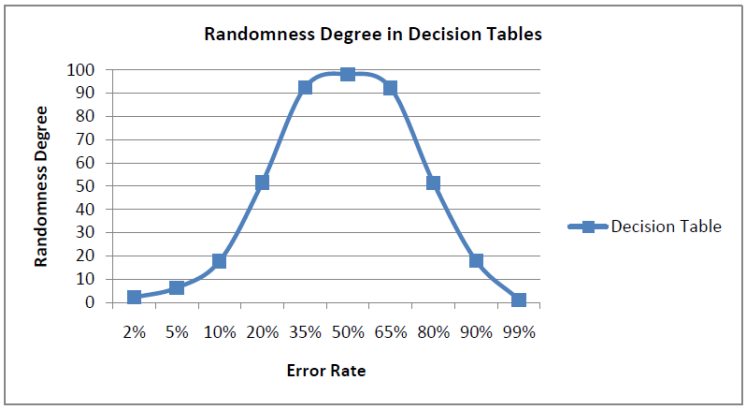

Figure 7.25: Randomness Degree in Decision Tables

rules method provides accurate assessment and enables users to determine errant patterns and measure their impact to the data quality.

7.6.2 Randomness Degree

7.6.2.1 Results and Discussions

The experiments for randomness degree are performed on ten synthetic binary (0,1) datasets where the error values are random and gradually increase. Figure 7.25 shows the randomness degree for these ten datasets. For each dataset, there is a correspondent D_T decision table that represents the error patterns found in the dataset. These patterns are used to compute the randomness degree for error (outlier) data in the dataset as in Figure 7.25.

As can be seen in Figure 7.25, the numbers of defective patterns increase gradually among 2%,5%,10%,20% and 35%. This increase in error rate contributes to the increase in the randomness degree from 2.1% to up 92.4%. Conversely, the randomness degree decreases when the distribution of errors is greater than 50%.Put alternatively, the randomness degree reaches its maximum level when

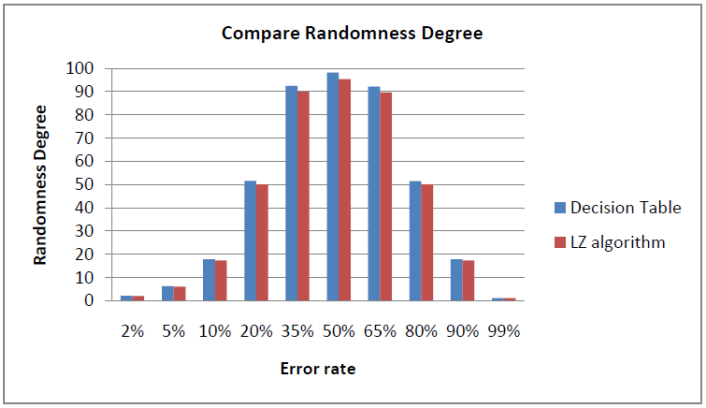

Figure 7.26: Compare Randomness Degree

the error rate is 50%.

To certify the validity of the experimental results, the study has rigorously compared the results obtained from the proposed randomness measurement with those from a well know algorithm called the LZ algorithm, as presented by Fisher et al. [2009]; Lempel and Ziv [1976]. The results for the proposed randomness measurements are based on the DT and LZ algorithm for the ten datasets with the following different error distribution: 2%,5%,10%,20%,35%,50%,65%,80%,90%,99% illustrate in Figure 7.26. The results are illustrated in Figure 7.26 and conclusively demonstrate that the decision table method can be used for measuring the randomness degree for poor data. Although, both algorithms show the same results for randomness measurement, the decision table method does not have problems caused by complexity associated with time like the LZ algorithm.

Another signification contribution of this approach is that users can distinguish defective patterns that tend to have a systematic attitude from random ones; see Table 7.10. After we measure the randomness degree RD_A, we classify defective errant patterns into two groups, systematic and random. Then,

Table 7.10: Distinguish between Systematic and Randomness Distribution

Error Rate	*RD*	*Number of*	*Systematic Patterns*		*Improve*	*Average Patterns*		*Improve*	*Patterns*
(%)	*(%)*	*DP*	*N. Patterns*	*Support*	*quality (%)*	*N. Patterns*	*Support*	*quality (%)*	*Impact quality*
2	*2.1*	*216*	20	2618	83.6	0	0	0	S
5	*6.2*	*644*	103	4998	81.1	140	390	6.3	S
10	*17.7*	*8698*	196	6103	71.1	105	445	5.1	S
20	51.5	10096	734	4835	47.9	4562	5261	52.1	S/R
35	*92.5*	*9505*	84	279	2.7	9421	9996	97.3	R
50	*98.2*	*10094*	3	9	0.088	10091	10272	99.91	R
65	*92.2*	*9480*	89	292	2.9	9391	9989	97.2	R
80	*51.5*	*5294*	747	5018	48.8	4547	5263	51.2	R/S
90	*17.8*	*1831*	178	7577	73.7	41	214	2.1	S

we calculate the impact of systematic and random errant patterns on quality by dividing the total support of systematic or random patterns to the total support of all errant patterns. This enables users to determine which type of errors (systematic (S) or random (R) have the most impact on the quality and thereby determine potential patterns for improving data quality, Table 7.10.

Table 7.11: Condition Table

		Condition							
		Size		*Useful Patterns*			*Less Useful Patterns*		
Error (%)	*#Rp in DT_A*	*#Pattern in Condition*	*Total (Sup)*	*#Significant Pattern*	*Total (Sup)*	*Improve quality (%)*	*#Insignificant Pattern*	*Total (Sup)*	*Improve quality (%)*
20	4562	224	4381	81	3603	82.24	143	778	17.75
35	9421	503	9281	184	6431	96.29	319	2850	30.70
50	10091	255	10045	138	6055	60.27	117	3990	39.72
65	9391	951	9391	395	7212	76.80	556	2179	23.20
80	4547	210	4547	59	3455	75.98	151	1092	24.02

Table 7.12: Decision Table

		Decision							
		Size		*Useful Patterns*			*Less Useful Patterns*		
Error (%)	*#Rp in DT_A*	*#Pattern in Decision*	*Total (Sup)*	*#Significant Pattern*	*Total (Sup)*	*Improve quality (%)*	*#Insignificant Pattern*	*Total (Sup)*	*Improve quality (%)*
20	4562	588	4489	197	3539	78.83	390	950	21.16
35	9421	508	9278	181	6532	70.4	325	2746	29.59
50	10091	1023	10079	658	7641	75.82	365	2438	24.18
65	9391	255	9387	94	6391	68.08	161	2996	31.91
80	4547	624	4547	196	3507	77.13	428	1040	22.87

Random distribution of error data has a linear distribution and usually produces a larger numbers of patterns. By analysing random patterns, users can

observe that these errant random patterns are not totally distinct from each other. Hence, we have introduced granule taxonomy to solve this problem. In this experiment, only, one level from DT_A guarantees significant results when exposing a small number of useful errant patterns for data quality purposes. We analyse random patterns for error rates of 20%,35%,50%,65%,80% because these error rates produce very high random errors. For example, a table with an error rate of 20% as shown in Table 7.10, produces a total of 5296 errant patterns; the errant random patterns from an error rate of 20% is 4562 patterns.

By contracting granule taxonomy to only one level, the algorithm returns a very small number of patterns with 224 and 588 defective patterns for condition and decision tables respectively. Applying this algorithm enables users to define systematic and random errant patterns which we can classify as useful for solving severe quality problems, and less severe errant patterns which we can classify as less useful patterns for assessing data quality. Referring back to 20% in the condition table, the number of useful errant patterns is 81 and the number of less useful errant patterns is 143 (see Table 7.11).

Similarly in the case of the decision table,Table 7.12, the useful and less useful patterns for data quality are 197 and 390 patterns respectively. Considering only the useful 81 errant pattern in 7.11 and 197 errant patterns in Table 7.12, the users become aware that these errant patterns could improve or impact on the data quality by 82.24 for the condition table, Table7.11 and 78.83 for the decision table, 7.12 instead of 4562 errant random patterns from level 0 in DT_A in Table 7.10.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Data quality has become a critical issue, drawing the attention of the many organisations that currently rely on the expanding area of database technologies for storing massive amounts of data, retrieving relevant information, and affording access to the heterogeneous resources. The capacities offered by these technologies are very advantageous to firms, as they can facilitate efficient and effective operational processes that positively reflect on the organisation and also on customer satisfaction. However, data quality is a major challenge affecting proper delivery of these benefits. The quality of the organisational information systems is impacted by the emergency of various poor data quality dimensions, particularly the dimension of outlier data. The appearance of outliers in a database or data warehouse can have a severe impact on operational costs, the decision making process, customer satisfaction and the accuracy of data mining projects.

Many critical applications are increasingly highly dependent on effective outlier solutions for maintaining consistent database and data warehouses, detecting fraudulent activities, exposing variances, improving the accuracy of the analysis

and also data mining projects. Hence, this thesis has thoroughly and deeply investigated and sought to uncover the critical issues that prevent a movement towards automated data quality for outlier data. Motivated by the issues associated with outlier data, this thesis has proposed a number of distinctive contributions:

- **Extracting Candidate Patterns**. The challenge for data quality research when detecting outlier data is that users need to mine large dimensional spaces in order to expose outlier data. The proposed Extracting Candidate Patterns minimise and approximate the location of outlier data in a relatively small size dataset. To minimise the mining space, this thesis has utilised new approaches to identify frequent patterns based on RST. Unlike frequent pattern mining algorithms, which are based on apriori algorithms, the DT and WDT algorithms do not make multiple passed over the dataset, as they are concerned with maximal patterns. Hence, the number of patterns or granules found in the DT and WDT is much smaller and more manageable than the number of the patterns found by frequent pattern mining algorithm. By reducing and approximating the number of candidate outlier patterns or granules based on the DT or WDT, it is easy to introduce an effective solution for mining outlier data, as the size of outlier sets is small.

- **Outlier Detection Algorithms**. These outlier algorithms enable users to effectively detect outlier data in categorical and mixed attributes datasets. The thesis introduced three outlier algorithms. (1) The GBOD algorithm introduced the weighted discernibility matrix. The experiments on this algorithm proved that the new weighted discernibility matrix introduced in

this thesis is effective for mining outlier data. (2) The RWDT algorithm utilised the WDT algorithms to classify patterns into three groups of outlier patterns: uncertain pattern and frequent patterns. The RWDT does not compute the distance between granules like the GBOD algorithm and therefore is more efficient for mining large datasets. (3) The CG algorithm. The CG algorithm eliminated the problems associated with specifying a number of nearest neighbours and the minimum distance, as the CG algorithm represents all granules in WDT in one single granule, which is used to determine the distance from other candidate granules L_G.

- **Quality Assessment**. To provide continuous automated solutions for poor data, users need to frequently assess the quality of their information systems in order to determine any quality changes or to study the behaviour of poor data. This thesis understands the essential elements of quality assessment and therefore introduces two essential techniques for assessing the quality of data and measuring the degree of randomness in poor data such as outlier. This has great advantages when measuring and specifying the locations of the most severe data errors, by determining proper techniques and resources to deal with poor data, enabling decision makers to ascertain whether the process used by their information systems requires re-engineering, and also assessing the effectiveness of the solutions used in capturing and preventing poor data from gaining access to the systems.

This thesis is unlike other existing work as it investigates the dimensionalities of the problem of data quality research, to achieve understanding. The focus of previous work has been on exposing poor data, regardless of whether it is outlier,

inaccurate, or incomplete data. The great advantage of this thesis is that it has looked at problems from the perspective of ensuring a continuous automated quality improvement. Hence, the proposed prototype systems facilitates this goal of automated quality improvement as the thesis not only detects poor data (outlier) but also assesses quality change and the allocation of the severity of outlier data.

Another distinct significant component of the thesis is that outlier algorithms do not require users' involvement in specifying a number of parameters or thresholds (such as the number of nearest neighbours or a minimum support threshold). Hence, the outlier results found by employing the proposed algorithms are not sensitive to setting parameters and therefore provide more accurate outlier detection compared to state-of-the-art algorithms.

The thesis conducted extensive experimental evaluation of several real and synthetic datasets. The results shown in the experimental studies for both outlier detection and quality assessment are also promising. The thesis firstly conducted several experiments to evaluate the effectiveness of the proposed three outlier algorithms GBOD, RWDT and CG in comparison to state-of-the-art algorithms. The comparison results indicated the significance of the proposed solutions for finding outliers.

Additionally, the thesis studied the quality change that occurs with noisy data. This enables users to assess any quality changes in a data set. The first proposed algorithm related to quality assessment in this thesis is the Decision rule algorithm. With a decision rule algorithm, users can exposes the patterns of defective data and measure its severity. Hence, users can precisely know which patterns that have the most impact on data quality. The thesis also considers

that errors are systematic and randomly appear across rows and columns. The proposed randomness measurement of error data enables users to allocate the location of the most severely affected data.

8.2 Limitations

The thesis has some limitations that have not been investigated. These limitations are beyond the scope of this work.

- The thesis limited its scope to two data types: categorical and mix attribute datasets. Thus, the proposed outlier algorithms are not applicable when seeking to handle outlier data in numeric datasets.
- The focus of the thesis was on improving the effectiveness of outlier detection, as represented by the introduction of a new direction for mining outlier data. The efficiency of the proposed algorithms are not considered and therefore the datasets used in the experiments are not very large dimensional datasets.

8.3 Future Work

There are several interesting directions that this study could be extended to follow in the future. The first of these would be to improve the WDT to enable users to contract a granule tree. The proper implementation of a granule tree could make a significant contribution for mining and predicting subspace outliers, and therefore improve efficiency when mining outlier data.

Another direction involves improving on the ideas proposed to support quality improvement. In a quality improvement task, there are two approaches: error correction and error prevention. For the purpose of error correction, the proposed algorithms, particularly the RWDT for attribute outlier detection, could be improved to detect noise values and re-correct them. The second quality improvement solution prevents such defective errors from accessing databases and data warehouses, by extracting rules from any of the proposed algorithms.

The ideas in this thesis can be extended to cover other data quality dimensions, including incomplete data, incorrect data and duplication. The ultimate benefits of these contributions would be to introduce effective systems that can handle quality problems in a very automated way.

References

C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD Rec.*, 30:37–46, May 2001. ISSN 0163-5808. 45, 121

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, volume 1215, pages 487–499–, Santiage, Chile., 1994. Citeseer. 37, 48

R. Agrawal, T. Imieliski, and A. Swami. Mining association rules between sets of items in large databases. 22(2):207–216–, 1993. 34, 35, 37, 64

N. Alkharboush and Y. Li. A decision rule method for data quality assessment. In *in Proceedings of the 15th International Conference on Information*, volume 3, pages 84–95, Little Rock, USA, 2010. ACM. 109

P.D. Allison. *Missing data*, volume 136. Sage Publications, Inc, 2001. 56

N. B. Amanda and K. E. Craig. An introduction to modern missing data analyses. *Journal of School Psychology*, 48(1):5 – 37, 2010. ISSN 0022-4405. doi: DOI: 10.1016/j.jsp.2009.10.001. 56

F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining*

and Knowledge Discovery, PKDD '02, pages 15–26, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44037-2. 46

D. Ballou, R. Wang, H. Pazer, and G.K. Tayi. Modeling information manufacturing systems to determine information product quality. 44(4):462–484–, 1998. ISSN 00251909. 21

D. P. Ballou and H. L. Pazer. Modeling data and process quality in multi-input, multi-output information systems. *Management Science*, 31(2):150–162–, 1985. ISSN 00251909. 2, 20, 101

D. P. Ballou and G.K. Tayi. Enhancing data quality in data warehouse environments. *Communications of the ACM*, 42(1):73–78–, 1999. ISSN 0001-0782. 100

V. Barnett. The study of outliers: Purpose and model. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 27(3):pp. 242–250, 1978. 45

C Batini and M Scannapieco. *Data quality: Concepts, methodologies and techniques*. Springer-Verlag New York Inc, 2006. xii, 51, 54, 100

C. Batini, C. Cappiello, C.. Francalanci, and A. Maurino. Methodologies for data quality assessment and improvement. 41(3):1–52–, 2009. ISSN 0360-0300. 3

S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 29–38, New York, NY, USA, 2003. ACM. 46, 125

R.J. Bayardo Jr. Efficiently mining long patterns from databases. In *ACM Sigmod Record*, volume 27, pages 85–93. ACM, 1998. 36

N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The r*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19:322–331, May 1990. 46

L. Berti-Equille. Data quality awareness: a case study for cost optimal association rule mining. 11(2):191–215–, 2007. 30

S. Besiki, G. Les, B. T. Michael, and C. S. Linda. A framework for information quality assessment. 58(12):1720–1733–, 2007. ISSN 1532-2890. 3

D Bitton, J Millman, and S Torgersen. A feasibility and performance study of dependency inference. pages 641–. IEEE Computer Society, 1989. 59

P. Bohannon, F. Wenfei, F. Geerts, J. Xibei, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 746–755–, 2007. 61, 64

M. M. Breunig, H. Kriegel, Raymond T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. ACM. 44, 47

C.E. Brodley, M.A. Friedl, et al. Identifying and eliminating mislabeled training instances. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 799–805, 1996. 44

T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. *Principles of Data Mining and Knowledge Discovery*, pages 1–42, 2002. 37, 38

T. Calders and B. Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206, 2007. 37, 38

C. Cappiello, C. Francalanci, and B. Pernici. A self-monitoring system to satisfy data quality requirements. 3761:1535–, 2005. 50

S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. 26(1):65–74–, 1997. ISSN 0163-5808. 28, 29

C.Y. Chen, S.C. Hwang, and Y.J. Oyang. Analysis and summarization of correlations in data cubes and its application in microarray data analysis. 9(1): 43–57–, 2005. 38

Y. Chen, D. Miao, and R. Wang. Outlier detection based on granular computing. In *Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing*, RSCTC '08, pages 283–292, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88423-1. doi: 10.1007/978-3-540-88425-5_29. 49

T. Dasu and T. Johnson. *Exploratory data mining and data cleaning*. Wiley, New York, 2003. 4, 22, 54

W. Eckerson. Data warehousing special report: Data quality and the bottom line. *Applications Development Trends May*, 2002. 2

R. Elmasri and S. Navathe. *Fundamentals of Databases Systems*. Addison-Wesley, Boston, 2007. 27, 28

C.K. Enders. *Applied missing data analysis*. The Guilford Press, 2010. 56

A. Even and G. Shankaranarayanan. Dual assessment of data quality in customer databases. *Journal of Data and Information Quality*, 1(3):1–29–, 2009. ISSN 1936-1955. 54

W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. 33(2):1–48–, 2008. ISSN 0362-5915. 58

U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996. xii, 31, 32

C. Fei and J. M. Ren. Discovering data quality rules. 1(1):1166–1177–, 2008. ISSN 2150-8097. xii, 58, 59, 61, 62, 63

L. Feng, J. X. Yu, H. Lu, and J. Han. A template model for multidimensional inter-transactional association rules. 11(2):153–175–, 2002. ISSN 1066-8888. 34

W. C. Fisher, J. M. E. Lauria, and C. C. Matheus. An accuracy metric: Percentages, randomness, and probabilities. *Journal of Data and Information Quality*, 1(3):1–21–, 2009. ISSN 1936-1955. xii, 56, 57, 102, 108, 153

Y. Ge, J. Ruoming, and G. Agrawal. Impact of data distribution, level of parallelism, and communication frequency on parallel data cube construction. In *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, pages 8 pp.–, 2003. 29

A. Ghoting, S. Parthasarathy, and M.E. Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008. 46

L. Golab, H. Karloff, F. Korn, D. Srivastava, and B. Yu. On generating near-optimal tableaux for conditional functional dependencies. 1(1):376–390–, 2008. 60, 61, 62

J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. pages 420–431–. Citeseer, 1995. 34

J. Han and Y. Fu. Mining multiple-level association rules in large databases. *Knowledge and Data Engineering, IEEE Transactions on*, 11(5):798–805, 1999. 34

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, London, 2001. 29, 33, 34, 38

J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, pages 1–12. ACM, 2000. 36

J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8:53–87, 2004. ISSN 1384-5810. doi: 10.1023/B:DAMI.0000005258.31418.83. 36

D.M. Hawkins. *Identification of outliers*. Chapman & Hall, 1980. 9, 44, 45

S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier detection using replicator

neural networks. *Data Warehousing and Knowledge Discovery*, pages 113–123, 2002. 121

Z. He, X. Xu, Z.J. Huang, and S. Deng. Fp-outlier: frequent pattern based outlier detection. *Computer Science and Information Systems/ComSIS*, 2(1):103–118, 2005. 44, 48, 125, 130

Z. He, S. Deng, X. Xu, and J. Huang. A fast greedy algorithm for outlier mining. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, pages 567–576, Berlin, Heidelberg, 2006. Springer-Verlag. 48, 125, 130

J. Hipp, U. Guntzer, and U. Grimmer. Data quality mining - making a virtue of necessity. In *Workshop on Research Issues in Data Mining and Knowledge Discovery DMKD*, pages –, Santa Barbara, CA, USA,, 2001. 37

K. Huang, Y. W. Lee, and R. Y. Wang. *Quality information and knowledge*. Prentice Hall PTR, 1999. 53

Y. Huhtala, J. karkkainen, P. Porkka, and H. Toivonen. Efficient discovery of functional and approximate dependencies using partitions. pages 392–401–. Citeseer, 1998. 60

Y. Huhtala, J. Karkkainen, P. Porkka, and H. Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. 42(2): 100–, 1999. 59, 60, 61

F. Jiang, Y. Sui, and C. Cao. Outlier detection using rough set theory. *Rough*

Sets, Fuzzy Sets, Data Mining, and Granular Computing, pages 79–87, 2005. 49

F. Jiang, Y. Sui, and C. Cao. Some issues about outlier detection in rough set theory. *Expert Systems with Applications*, 36(3):4680–4687, 2009. 49

W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 293–298, New York, NY, USA, 2001. ACM. 47

W. Jin, A. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, pages 577–593, Berlin, Heidelberg, 2006. Springer-Verlag. 47

T. Johnson, I. Kwok, and R. Ng. Fast computation of 2-dimensional depth contours. In *Proceedings of the 4th ACM SIGKDD international conference on Knowledge discovery and data mining*, volume 1998, pages 224–228, 1998. 45

W. Kim, B. J. Choi, E. K. Hong, S. K. Kim, and D. Lee. A taxonomy of dirty data. 7(1):81–99–, 2003. ISSN 1384-5810. 24

E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 392–403, San Francisco, CA, USA, 1998. 10, 44, 45, 46

E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 211–222, San Francisco, CA, USA, 1999. 9, 10, 44, 45, 46

A. Koufakou, E.G. Ortiz, M. Georgiopoulos, G.C. Anagnostopoulos, and K.M. Reynolds. A scalable and efficient outlier detection strategy for categorical data. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 2, pages 210 –217, oct. 2007. doi: 10.1109/ICTAI.2007.125. 48, 125, 130

A. Koufakou, M. Georgiopoulos, and G. Anagnostopoulos. Detecting outliers in high-dimensional datasets with mixed attributes. In *2008 International Conference on Data Mining DMIN*, 2008. 49, 50

A. Koufakou, J. Secretan, and M. Georgiopoulos. Non-derivable itemsets for fast outlier detection in large high-dimensional categorical data. *Knowledge and Information Systems*, 29:697–725, 2011. ISSN 0219-1377. 10.1007/s10115-010-0343-7. 48

A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 157–166, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081891. 121

Y. W. Lee, D. M. Strong, B.K. Kahn, and R. Y. Wang. Aimq: a methodology for information quality assessment. *Information & Management*, 40(2):133–146–, 2002. ISSN 0378-7206. 2, 3, 21, 22, 52, 101

A. Lempel and J. Ziv. On the complexity of finite sequences. *Information Theory, IEEE Transactions on*, 22(1):75–81–, 1976. ISSN 0018-9448. 153

Y. Li. Interpretations of discovered knowledge in multidimensional databases. In *Proceedings in IEEE International Conference on Granular Computing*, page 307, 2007. 39, 74

Y. Li and KD Joshi. Data cleansing decisions: Insights from discrete-event simulations of firm resources and data quality. *Journal of Organizational Computing and Electronic Commerce*, 22(4):361–393, 2012. 1

Y.. Li and N. Zhong. Interpretations of association rules by granular computing. In *Proceedings of Third IEEE International Conference on Data Mining*, pages 593 – 596, 2003. 39, 74

R. J. A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley, New York, 2002. 55, 56

B. Liu, J. Yin, C. Xiao, L. Cao, and P.S. Yu. Exploiting local data uncertainty to boost global outlier detection. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 304 –313, dec. 2010. doi: 10.1109/ICDM.2010.10. 47

S Lopes, JM Petit, and L Lakhal. Efficient discovery of functional dependencies and armstrong relations. pages 350–364–, 2000. 60

S.. Madnick, R. Wang, Y. Lee, and H. Zhu. Overview and framework for data and information quality research. 1(1):1–22–, 2009. ISSN 1936-1955. 3

H. Mannila and K.J. Raiha. On the complexity of inferring functional dependencies. 40(2):237–243–, 1992. 59

A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 169–178, New York, NY, USA, 2000. ACM. 46, 47

R.B. Messaoud, S. L. Rabaseda, O. Boussaid, and R. Missaoui. Enhanced mining of association rules from data cubes, 2006. xii, 30, 38

M. Otey, A. Ghoting, and S. Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, 12:203–228, 2006. ISSN 1384-5810. 10.1007/s10618-005-0014-6. 44, 48, 49, 50, 119, 125, 130

S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. Loci: fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315 – 326, march 2003. 47

Z. Pawlak. *Rough sets: Theoretical aspects of reasoning about data*. Springer, Kluwer, Dordrecht, 1991. 38, 103

Z. Pawlak. In pursuit of patterns in data reasoning from data-the rough set way. In *Proceedings 3rd International Conference on Rough Sets and Current Trends in Computing*, pages 949–957, USA, 2002. Springer. 39

Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Information Sciences*, 177(1):41–73, 2007a. ISSN 0020-0255. 39

Z. Pawlak and A. Skowron. Rudiments of rough sets. *Information Sciences*, 177 (1):3–27, 2007b. 39, 74

J. Pei and J. Han. Constrained frequent pattern mining: a pattern-growth view. *ACM SIGKDD Explorations Newsletter*, 4(1):31–39, 2002. 36

L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218–, 2002. 53, 54

E. Rahm and H. Do. Data cleaning: Problems and current approaches. 23(4): 3–13–, 2000. 25, 59

S. Ramaswamy, R Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29:427–438, May 2000. 9, 46, 47

C. Rauszer and A. Skowron. The discernibility matrices and functions in information systems. *Intelligent decision support. Handbook of applications and advances in the rough set theory. Kluwer, Dordrecht*, pages 331–362, 1992. 85

T. C. Redman. *Data quality for the information age*. Artech House Boston, MA, 1996. 2, 21, 22, 54, 101

T. C. Redman. The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2):79–82–, 1998. 2, 20, 50, 100

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):pp. 581–592, 1976. ISSN 00063444. 55

I. Ruts and P. J. Rousseeuw. Computing depth contours of bivariate point clouds. *Comput. Stat. Data Anal.*, 23:153–168, November 1996. 45

I. Savnik and P.A. Flach. Discovery of multivalued dependencies from relations. 4(3):195–211–, 2000. 60, 61

J.L. Schafer and J.W. Graham. Missing data: Our view of the state of the art. *Psychological methods*, 7(2):147, 2002. 56

Z. Shichao, Z. Jilian, Z. Xiaofeng, and H. Zifang. Identifying follow-correlation itemset-pairs. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 765–774–, 2006. 34

F. Silvers. *Building and Maintaining a Data Warehouse*. Auerbach, Hoboken, 2008. 4, 22, 28

A. Skowron and P. Synak. Reasoning in information maps. *Fundamenta Informaticae*, 59(2-3):241–260, 2004. 85

D. M. Strong, Y. W. Lee, and R.Y. Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110–, 1997. ISSN 0001-0782. 2, 22, 101

J. Tang, Z. Chen, A. Fu, and D. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, pages 535–548, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43704-5. 47

S. Tsumoto and S. Hirano. Visualization of rule's similarity using multidimen-

sional scaling. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 339–346–, 2003. 34

A. K. H. Tung, L. Hongjun, H. Jiawei, and F. Ling. Efficient mining of inter-transaction association rules. 15(1):43–56–, 2003. ISSN 1041-4347. 34

P. Vassiliadis, M. Bouzeghoub, and C. Quix. Towards quality-oriented data warehouse usage and evolution. 25(2):89–115–, 2000. ISSN 0306-4379. 28

P. Vassiliadis, Z. Vagena, S. Skiadopoulos, N. Karayannidis, and T. Sellis. Arktos: towards the modeling, design, control and execution of etl processes. 26(8): 537–561–, 2001. 28

Y. Wand and R.Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95–, 1996. ISSN 0001-0782. 2, 20, 101

R. W. Wang and D.M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–33–, 1996. ISSN 07421222. 2, 22, 52, 101

R. Y. Wang and H.B. Kon. Toward total data quality management (tdqm). In *Information technology in action: trends and perspectives*, pages 179–197–. Prentice-Hall, Inc., 1993. 21, 52

R. Y. Wang, V. C. Storey, and C. P. Firth. A framework for analysis of data quality research. 7(4):623–640–, 1995. 1, 21

F. Wenfei, F. Geerts, L. Lakshmanan, and X. Ming. Discovering conditional

functional dependencies. In *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, pages 1231–1234–, 2009. 59

C. Wyss, C. Giannella, and E. Robertson. Fastfds: A heuristic-driven, depth-first algorithm for mining functional dependencies from relation instances extended abstract. pages 101–110–, 2001. 59, 60

H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *Knowledge and Data Engineering, IEEE Transactions on*, 18(3):304–319, 2006. 44

Y. Xu and Y. Li. Generating concise association rules, 2007. 38

X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 314–323. ACM, 2005. 33

W. Yang, Y. Li, J. Wu, and Y. Xu. Granule mining oriented data warehousing model for representations of multidimensional association rules. *International Journal of Intelligent Information and Database Systems*, 2(1):125–145, 2008. 39, 74

J. X. Yu, W. Qian, and A. Lu, H.and Zhou. Finding centric local outliers in categorical/numerical spaces. *Knowl. Inf. Syst.*, 9(3):309–338, March 2006. 49

M.J. Zaki. Mining non-redundant association rules. *Data mining and knowledge discovery*, 9(3):223–248, 2004. 38

M.J. Zaki, C.J. Hsiao, et al. Charm: An efficient algorithm for closed association rule mining. Technical report, Citeseer, 1999. 36, 37, 38